Análisis Léxico sobre los *Tweets* de *Twitter*

Astrid Paola Bográn[1], José Luis Alonso Berrocal[1] y Luis Carlos García de Figuerola Paniagua[1]

[1]Departamento de Informática y Automática - Facultad de Traducción y Documentación, Universidad de Salamanca. Calle de Francisco Vitoria 6-16. 37008 Salamanca, España
{abogran,berrocal,figue}@usal.es

Resumen Este documento ofrece un acercamiento sobre el Análisis léxico enfocado en los *tweets* de *Twitter*, presentando el desarrollo de una aplicación web que se podrá conectar con *Twitter* involucrando el manejo de un clasificador de texto sobre la web, y de esta manera poder descubrir las características esenciales de los *tweets* seleccionados ya sea de forma individual o masiva, todo esto ejecutándose en tiempo real o bien agregando el contenido a la base de datos que ofrece la aplicación para posteriormente tratar los *tweets* al gusto del usuario. Durante el proceso de investigación se han utilizado técnicas de *stemming* y tokenización que ayudan a procesar el *tweet* de forma más limpia y sin ruido; igualmente, para la clasificación se han creado varios diccionarios en formato *XML* basados en las áreas de ciencia y tecnología, también diccionarios que ayuden a identificar palabras vacías; para realizar la clasificación se propuso el algoritmo Naïve Bayes.

Keywords: Análisis Léxico, Tweets, Twitter, Clasificación, Algoritmo Naïve Bayes

1. Introducción

En la actualidad las redes sociales han llegado a ser de gran importancia para la sociedad, es el caso de *Twitter* que actualmente cuenta con más de 500 millones de personas conectadas en tiempo real alrededor del mundo, gracias a esta red que más que un *microblogging* se ha convertido en una fuente de noticias al último minuto, los empresarios, famosos, investigadores, instituciones gubernamentales y privadas además de un sinfín de usuarios que también se conectan a esta red para compartir sus experiencias e inquietudes o noticias de última hora; con toda esta gran cantidad de información que se encuentra en la red y con su rápido crecimiento hacen que muchos quieran indagar un poco más sobre algunos perfiles que pudieran ser importantes para la investigación moderna y también ser parte de nuevos descubrimientos, soluciones o simplemente por encontrar su potencial dentro de la red. Pensamos entonces frente a tanta cantidad de datos, crear una herramienta que ayude a analizar lexicalmente el contenido presente en tiempo real, agregando diversas características para que el usuario pueda clasificar el contenido. Cuando hablamos de Análisis léxico se nos viene a

la cabeza una gran cantidad de palabras; de eso se trata, analizar todo el contenido que encontramos en una o varias cuentas especificas pertenecientes a la famosa red social *Twitter*. Para ello se ha creado una aplicación web la cual nos ayudará a descubrir las características de un *tweet*, permitiendo que las personas con diversos intereses o conocimientos la utilicen involucrando los patrones de sus propias áreas de trabajo.

La investigación se basa en la incorporación de técnicas de minería de texto web, utilizando el algoritmo Naïve Bayes para poder generar la clasificación y el Análisis sobre cada uno de los *tweets*. Estos podrán ser escogidos de manera individual o masiva. Entre otras técnicas empleadas dentro de la aplicación encontramos la utilización de *stemmings* y tokenización, los cuales son fundamentales a la hora de realizar la clasificación. Esto se logró con la utilización e implementación de varias herramientas de desarrollo como *Visual Studio 2010 .NET*, lenguajes de programación *C#*, *HTML5*, *JavaScript*, y para conseguir que la aplicación de desarrollo cliente se conecte con *Twitter* fue necesario la utilización de una Interfaz de programación de aplicaciones (API) la cual es la encargada del conjunto de llamadas sobre las bibliotecas para este caso se utilizó la librería *Linqto*Twitter, en la clasificación de los *tweets* fue preciso incorporar el servidor *UClasify*. Igualmente para obtener el acceso a los recursos protegidos de *Twitter* se utilizó el protocolo abierto para desarrolladores *OAuth*.

Sin todo este estudio de fondo no hubiera sido posible la obtención de los resultados y descubrimientos que se puedan realizar sobre alguna cuenta específica. El desarrollo de un sitio web como este supone, hoy en día, un avance tecnológico de alta calidad y a gran escala con expectativas futuristas referentes al manejo del contenido web pero específicamente sobre las redes sociales tales como *Twitter*.

2. Conocimiento Previo

Las Redes Sociales son básicamente unidades virtuales, donde existen usuarios que interactúan unos con otros a través de internet. Se agregan perfiles que tienen algo en común y la plataforma que se utiliza permite conectar gente que se conoce realmente o que desea conocerse por algún atributo que encuentren en común, según sea el caso de la plataforma en algunas se pueden compartir fotos, vídeos, centralizar recursos, foros de discusión, mensajes, etc [1]. Existen actualmente muchos estudios sobre las redes sociales, que se basan en gran medida en estudios individuales sobre el comportamiento de las personas, también están los enfocados en mecanismos que utiliza la red para unir todo este ambiente, donde analizan y describen los niveles de la red social.

Twitter se refiere a una red de información en tiempo real como servicio de *microblogging* que conecta con las últimas historias, ideas, opiniones y noticias sobre lo que las personas encuentran interesante. Fue creado por Jack Dorsey,

quien dispuso que la longitud máxima de cada mensaje fuera de 140 caracteres.

Twitter permite a la gente a seguir y comunicarse con los demás. Con el tiempo, esto ha demostrado ser un medio de comunicación de gran alcance debido a su funcionamiento y simplicidad. Actualmente *Twitter* cuenta con 500 millones de usuarios esto para el 2012, según un estudio realizado por *semiocast* [2]. Se ha convertido en uno de los principales canales de comunicación e información a nivel mundial, se calcula que alrededor de 400 millones de *tweets* son enviados cada día. Hoy en día existen muchos estudios sobre las redes sociales, especialmente centrados en el comportamiento de las personas, en la mayoría de los casos basados en los sentimientos, en donde también se utilizan técnicas de minería de texto utilizando una serie de algoritmos que ayudan a determinar ciertos patrones en el comportamiento de los humanos, aquí encontramos el caso del algoritmo Naïve Bayes que ayuda en la clasificación.

3. Análisis léxico

Se refiere al proceso de convertir una serie de caracteres en una secuencia de *tokens* agrupados que se leen de izquierda a derecha. El programa que realiza esta función se llama analizador léxico [3]. Para una clasificación automática de texto es importante recalcar que el Análisis léxico ayudará a extraer características que detallarán a cada documento, así como, sus clases o categorías. También en la definición de reglas que por lo general consisten en expresiones regulares, y definen el conjunto de posibles secuencias de caracteres que se utilizan para formar los *tokens* o los lexemas individuales [3].

La secuencia de caracteres que forman un *token* se llama lexema. El *token* es el símbolo general a la que un lexema pertenece. Existen dos métodos principales principales para el Análisis de una palabra:

- ***Tokens*:** Son secuencias de caracteres con un significado colectivo, [3] el proceso de formación de los *tokens* que permite el flujo de entrada de caracteres se llama tokenización, y el analizador léxico les clasifica de acuerdo con un tipo de símbolo.
- ***Stemming:*** Es un procedimiento que se utiliza para reducir una palabra a su raíz. Existen hoy en día diversos algoritmos que realizan *stemming*, como el caso del *Snowball*. Un *stem* (lema) es la porción que queda de una palabra después de retirar los afijos (es decir, los prefijos y los sufijos). Consiste en convertir todas las palabras parecidas a una forma común [4]. Entre algunas ventajas que proporciona el *stemming* es que reduce el tamaño del índice ya que el número de palabras es reducido y las variantes de la palabra se reducen a un concepto común eso mejora la importancia de la recuperación de información.

3.1. Técnicas y Librerias

API de *Twitter* 1.1. Ya que la información que fluye a través de *Twitter* es muy valiosa especialmente para nuevas investigaciones sobre el comportamiento

o descubrimiento de nuevos patrones que siguen los usuarios, uno de los objetivos de *Twitter* es lograr un equilibrio entre el fomento de desarrollo interesante y protección de los derechos tanto de *Twitter* y los usuarios. Para conceder un ambiente de desarrollo recto ha creado ciertas características para la utilización de esta tecnología en su versión 1.1 encontramos los siguientes puntos de interés:

- El usuario tiene 15 minutos después de la última ejecución dentro de *Twitter*, permitiendo 15 solicitudes por cada ventana. También entre otras convocatorias utilizadas recibirá un impulso de 180 solicitudes por ventana.
- Un requerimiento básico para todas las aplicaciones es la autenticación sobre todas sus peticiones con *OAuth* 1.0. Esto no sólo permite la visibilidad a prevenir conductas abusivas, sino que también nos ayudará a comprender mejor cómo las categorías de aplicaciones están utilizando la API. Este es un conocimiento para satisfacer mejor las necesidades de los desarrolladores a medida que se continúe evolucionando la plataforma. También toda la autenticación requiere contexto de usuario.
- Las aplicaciones cliente tienen un límite simbólico de 100,000 usuarios por *token*.

***Linqto*Twitter.** El proveedor *LINQ* sus siglas *Language Integrated Query* utiliza en su sintaxis un estándar para las consultas e incluye llamadas a métodos de cambios a través de la API de *Twitter*, lo ha implementado para su conexión con su API, se utiliza la sintaxis de *LINQ* estándar para las consultas e incluye llamadas a métodos de cambios a través de la API de *Twitter*. Entre las plataformas que soporta se incluyen *.NET* 3.5, *.NET* 4.0, *.NET* 4.5, *Silverlight* 4.0, *Windows Phone* 7.1, *Windows Phone* 8, *Window* 8 y *Windows Azure* [5].

Autenticación *OAuth*. *Open Authorization* es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite la autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web [6]. Fue lanzado hacia finales de 2007 y define un mecanismo para que una aplicación web (cliente) pueda acceder a la información de un usuario en otra (proveedor) sin tener que informar a la primera del usuario y contraseña.

Algoritmo Naïve Bayes. Es un clasificador probabilístico fundamentado en el teorema de Bayes, quien utilizó las matemáticas para encontrar la mejor clasificación y así descubrir el conocido teorema de la probabilidad [7]. Es un método importante no sólo porque ofrece un Análisis cualitativo de los atributos y valores que pueden intervenir en el problema, sino porque da cuenta también de la importancia cuantitativa de esos atributos. Su forma es simple y al mismo tiempo sorprendentemente eficaz es ampliamente utilizado en áreas como la clasificación de texto y filtrado de *spam*. Se han introducido una gran cantidad de modificaciones, por la estadística, minería de datos, aprendizaje automático, etc [8]. Un modelo bayesiano es fácil de construir, esto lo hace especialmente útil

para grandes bases de datos. Gracias a su simplicidad, este clasificador es bastante utilizado, especialmente en la recuperación de la información y manejo de texto en la web.

UClasify. El servidor Clasificación *uClassify* es impulsado por llamadas *XML* que permite a los usuarios crear y entrenar cualquier clasificador arbitrario. Se puede utilizar para categorizar las páginas web o cualquier otra tarea que requiera clasificación automática de texto. Está diseñado para manejar grandes cantidades de datos y puede procesar millones de documentos diarios [9]. El núcleo es un clasificador bayesiano Naïve con un par de pasos que mejora la clasificación. Muchos otros modelos sólo pueden responder Sí o No, pero no indican el grado. Los resultados que muestra este clasificador son probabilidades [0-1] [9] de un documento que pertenecen a cada clase. Esto es muy útil si se desea establecer un umbral para las clasificaciones. El uso de este modelo también hace que sea muy escalable en términos de tiempo de CPU para la clasificación.

Snowball. Es una librería construida en el lenguaje *Java*, en donde los algoritmos resultantes pueden representarse fácilmente. El compilador traduce a un *script* de *Snowball* (un archivo. Sbl) y dentro de él se crea un hilo seguro para subprocesos *ANSI* que puede ser ejecutado en *C* o un programa *Java*. Cada *script Snowball* produce un archivo y su cabecera correspondiente (con extensiones .c y .h) [10]. Se ha inventado en el lenguaje de *Snowball* reglas de algoritmos que contienen una serie de *stemmers* de lengua extranjera y vocabularios estándar de las palabras y sus equivalentes derivados donde se proporciona la raíz de cada palabra.

4. Trabajo Propuesto

Una vez conocidas las herramientas, librerías, algoritmos y términos relacionados con la lexicografía, podremos comenzar a crear una herramienta que propone el desarrollo de una aplicación que sea capaz de realizar un análisis léxico sobre los *tweets* de *Twitter*, en este caso para validar el sistema nos enfocaremos en la selección de los *tweets* basados en noticias relacionadas con ciencia y tecnología, a fin de efectuar sobre ellos diversos Análisis tanto cuantitativos como cualitativos. El elevado número de documentos de prensa disponibles hace inviable una selección manual, por lo que se decide aplicar un clasificador que seleccione de forma automática las noticias o documentos que traten de una u otra forma sobre temas relacionados en estas dos áreas. Asimismo se propone al usuario la creación de nuevas categorías descubriendo las características sobre algún perfil relacionado a una cuenta de *Twitter*.

4.1. Planteamiento del Problema

Pese a la gran cantidad de información y trabajos de investigación que encontramos en la web, hasta la fecha no se ha encontrado ningún analizador léxico

que ayude a clasificar las noticias en *Twitter*. Por esta razón se pretende diseñar y construir una aplicación que permita obtener todos los datos públicos que están involucrados en una cuenta; en este caso se proyecta recoger todos aquellos *tweets* que involucren las áreas de ciencia y tecnología, de forma masiva e individual, esto con el fin de poder procesar el *tweet* haciendo uso de tcnicas de *stemming*, tokenización, y clasificación del *tweet* descubriendo la correcta categorización entre las dos áreas. Se trata también de que el mismo usuario pueda crear y generar su propia clasificación dependiendo el área de su interés.

4.2. Objetivo General

Construir un analizador léxico que permita el ingreso a una cuenta de *Twitter* con el apoyo de diferentes técnicas que ayuden a identificar y clasificar *tweets* referentes a las áreas de ciencia y tecnología y que también el usuario pueda realizar las mismas operaciones de clasificación utilizando su propia clasificación.

4.3. Análisis del Sistema

El proceso que se pretende llevar a cabo describe un flujo comenzando desde lo que será el corazón o núcleo del sistema con la creación de un sitio web conectado a través de un *webservice* que responderá a las peticiones del sitio web cliente. Para la autenticación con *Twitter* se utilizó *OAuth* que hasta la fecha es la forma más común de autenticación de recursos en aplicaciones de *Twitter*, continuando con la iteración se decide utilizar el lenguaje de programación *C#* ya que es un lenguaje de programación independiente y de adecuación para escribir aplicaciones de cualquier tamaño; es el caso de este proyecto donde se desarrolla una aplicación web con la combinación de *Visual Studio .NET* 2010 utilizando el *framework* 4.0, la biblioteca *Linqto*Twitter y la API de *Twitter*. La Figura 1 describe el flujo de nuestro sistema.

Figura 1. Flujo del Sistema

Comenzando con el desarrollo del sistema nos enfocamos en el Análisis léxico basado en la clasificación de los *tweets* en las áreas de Ciencia y Tecnología; estableciendo parámetros a seguir para realizar cálculos y obtener algún descubrimiento sobre casos particulares. Nos centramos en las siguientes premisas:

1. Estudio sobre terminología utilizada en *Twitter*.
2. Análisis de entidades.
3. Observación sobre la recuperación de la información.
4. Visualización sobre los resultados obtenidos de los casos de estudio.

Se trabajará con cada una de las palabras contenidas dentro de un *tweet*, considerando como palabra una secuencia ininterrumpida de letras [11]. Se removerán las palabras contenidas en una lista estándar denominada palabras vacías para el español. El sistema realizara esta limpieza que es necesaria para la extracción de palabras de un *tweet*, lo que aplicaremos en nuestro Análisis y que se detalla a continuación:

- Eliminación de acentos.
- Conversión a minúsculas.
- Eliminación de cualquier carácter que no sea alfanumérico.

El texto resultante se someterá a un proceso de normalización o *stemming*, para el cual se debe trabajar con la librería basada en el *stemmer de SnowBall*; y como último punto también se deben eliminar aquellas palabras cuya longitud sea menor a cuatro caracteres.

4.4. Diseño del Sistema

El sistema consta de una serie de ventanas que básicamente están distribuidas en los siguientes 5 pasos principales:

Inicio. La ventana inicial describe la bienvenida y una breve explicación comentando el objetivo principal referente al Análisis léxico sobre los *tweets* de *Twitter*, indicando también que es una aplicación con un diseño sencillo y código entendible. Podemos ver esta ventana en la Figura 2.

Figura 2. Pantalla de Bienvenida

Autenticación. Este paso se realiza con la utilización de *OAuth* como mecanismo de autenticación para *Twitter* proporcionando un método para que los clientes accedan a los recursos del servidor en nombre del propietario (como un cliente diferente o un usuario final). Seguido de la autenticación podremos ver el resumen de la cuenta visualizando los resultados: nombre de usuario, total de amigos, seguidores, actualizaciones y favoritos.

Verificando la conexión a través del Habla. Una vez autenticados con la cuenta personal de *Twitter*, la aplicación se conecta a través del *web service* mostrando la pantalla para la prueba de la conexión con *Twitter* y para hacer la aplicación más dinámica se ha decidido trabajar con el habla.

Análisis léxico. Esta pantalla detalla una parte de nuestro objetivo principal. En ella encontraremos varias características que nos ayudaran a identificar el proceso que debemos seguir para realizar el Análisis léxico sobre un *tweet* que podremos seleccionar para posteriormente ser evaluado.

Clasificación y Trabajo con Diccionarios. Aquí se definen los diccionarios que han sido creados con la ayuda de la herramienta *wordnet* y posteriormente desarrollados en un formato *XML* para que puedan ser leídos por el clasificador y categorizar adecuadamente los *tweets* seleccionados. En esta pantalla encontramos básicamente un botón principal Clasificador que se encarga de clasificar los *tweets* de forma automática según las categorías ciencia y tecnología. Este botón depende de la selección que se haya hecho en la pantalla anterior, es decir que si se quieren clasificar los *tweets* de forma masiva se pueden utilizar los que se encuentran dentro de la base de datos o si se quiere clasificar en tiempo real mediante la conexión con *Twitter*.

Gestión de Tweets. Esta sección de la aplicación web cuenta con una serie de características donde se puede trabajar con diferentes cuentas de usuario aún sin estar autenticados donde podrán buscar los *tweets* sobre cualquier perfil que no sea privado.

En esta pantalla podremos ingresar la mención o el usuario de alguna cuenta y agregar la cantidad de *tweets* que desee visualizar sobre la cuenta solicitada. Además, la aplicación también permite guardar los *tweets* seleccionados en un archivo de texto plano o en directamente en la base de datos, y permite crear y entrenar un nuevo clasificador. Ver Figura 3.

4.5. Desarrollo del Sistema

A fin que el Análisis y el diseño del sistema funcionen correctamente, se definieron varios proyectos, donde cada uno de ellos contiene una funcionalidad específica, las cuales se detallarán en esta sección.

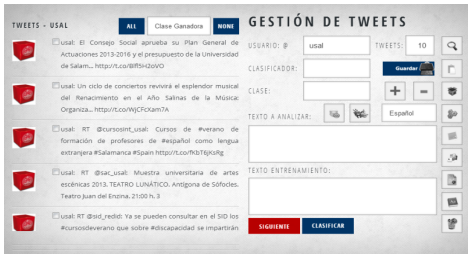

Figura 3. Resumen de cuenta

Conexión con *Twitter*. Como lo hemos mencionado anteriormente, la conexión con *Twitter* se realiza con la ayuda de las credenciales proporcionadas por *OAuth*. Esta conexión se realiza a través de las siguientes líneas de código (ver Listado 2.1)

Listado 2.1. Código conexión

```
static WebAuthorizer auth;
public static \emph{Twitter}Context obtenerContexto(ref WebAuthorizer _auth)
{
  const string OAuthCredentialsKey = "OAuthCredentialsKey";
  \emph{Twitter}Context twitterCtx;
  IOAuthCredentials credentials = new SessionStateCredentials();
  if (credentials.ConsumerKey == null || credentials.ConsumerSecret == null)
  {
     credentials.ConsumerKey =
          ConfigurationManager.AppSettings["twitterConsumerKey"];
     credentials.ConsumerSecret =
          ConfigurationManager.AppSettings["twitterConsumerSecret"];
  }
}
```

Consulta Sobre los Detalles de la Cuenta. La Tabla 1 describe los detalles que pueden ser obtenidos de una cuenta de *Twitter*.

Clasificación Masiva. La siguiente codificación (Listado 2.2) muestra la clasificación que se realiza masivamente en la pantalla Diccionarios de la aplicación web.

Listado 2.2. Clasificación Masiva de los Tweets

```
var listtweets =
from tweet in twitterCtx.Status
where tweet.Type == StatusType.Home && tweet.Count == 10
select tweet;

foreach (var tw in listtweets)
{
  string tweetText = limpiarCadena(tw.Text);
```

```
  List<uClassifyResponse> uRes =
       ClassificarTexto(tweetText,"TechnologyorScience", false, true);
  string classifyName = ((Math.Round(uRes[0].percentage, 2) >
       Math.Round(uRes[1].percentage, 2)) ? uRes[0].className:
       ((Math.Round(uRes[0].percentage, 2) < Math.Round(uRes[1].percentage,
       2) ? uRes[1].className : "None")));

  tweetList.Add(new Tweet(nombre: classifyName, imageURL:
       tw.User.ProfileImageUrl,
  ultimoTweet: limpiarCadena(tw.Text), seguidores:
       Convert.ToInt32(tw.User.FollowersCount),
  estados: Convert.ToInt32(tw.User.StatusesCount), siguiendo:
       tw.User.Following, retweeted:
       Convert.ToBoolean(tw.Retweeted)));
}
```

Trabajando con *Stemming*. Consiste en una biblioteca de clases que no es más que una dll para poder ser utilizada con *.NET*. Específicamente se ha desarrollado la traducción a *.NET* de la librería que está basada en el lenguaje *java*. Una vez que tenemos nuestra propia dll, podremos utilizarla específicamente para la limpieza de los *tweets* o el contenido que se pretenda analizar.

La librería cuenta con alrededor de quince idiomas diferentes entre ellos el inglés y español, que son básicamente con los que trabajaremos para realizar la clasificación de los emphTwitter ya que nuestros diccionarios están basados en estos dos lenguajes. Utilizamos el *Stemming* porque es uno de los métodos más precisos para agrupar palabras con un significado similar, que corresponde al caso de nuestros dos clasificadores basados en ciencia y tecnología.

Creación de Diccionarios. El segundo caso define otra biblioteca de clases que consiste en gestionar los diccionarios que se refieren a la definición de una librería para la creación y manipulación de los archivos en *XML* que son la clave fundamental en el proyecto donde están contenidas todas las palabras para la clasificación en las áreas de ciencia y tecnología basadas en los *tweets*. Los

Nombre	**Propósito**	**Tipo de Dato**	**Requerido**
EndSessionStatus	Respuesta de la solicitud para finalizar la sesión	String	No
RateLimitStatus	Clasificación del texto limitada	RateLimit Status	No
Settings	configuración de la cuenta	Settings	No
Totals	totales actuales	Totals	No
Type	Tipo de cuenta consulta	Account Type	No
User	Usuario devuelve la veracidad de las Credenciales de la Consultas	User	No

Tabla 1. Filtros/ Parámetros, detalles de la cuenta.

diccionarios han sido creados con la ayuda de *WordNet*, que contiene una gran base de datos léxica en el idioma Inglés donde encontramos sustantivos, verbos, adjetivos y adverbios que se agrupan en conjuntos de sinónimos cognitivos (*synsets*), cada uno expresando un concepto distinto. Los términos en *Synsets* están vinculados entre sí mediante relaciones conceptuales, semánticas y léxicas [12]. Gracias a esta herramienta se logró la extracción de las palabras relacionadas con ciencia y tecnología que a su vez han pasado por un proceso de traducción de forma manual, y para poder ser utilizadas en nuestro sistema se realizo la construcción de los diccionarios en formato *XML*.

5. Resultados

Los resultados están basados en el estudio de un total de 3690 *tweets* que han sido almacenados en la base de datos creada. Para comprobar que el algoritmo de clasificación de texto Naïve Bayes y las técnicas léxicas funcionan correctamente en el sistema, se decidió realizar dos estudios, uno sin utilizar la técnica de emphstemming y el segundo basándonos en esta técnica y al mismo tiempo utilizando todo el tratamiento de limpieza que realiza la aplicación.

Clasificación de *tweets*:

- Ciencia: 2050 *tweets*.
- Tecnología: 627 *tweets*.
- Otros considerados como ambiguos: 1013 *tweets*.

Prueba Sin Utilizar Stemming. De la clasificación sin utilizar las técnicas de Análisis léxico como el *stemming* o alguna otra técnica de limpieza sobre los *tweets*, el algoritmo Naïve bayes logra una exactitud de 52.14 %, y una precisión por clase con los siguientes valores (Tabla 2): En la Figura 4 podemos visualizar

	Ciencia	**Tecnología**	**Ninguno**	**Totales**
Ciencia	1133	604	313	2050
Tecnología	185	364	78	627
Ninguno	301	285	427	1013
Totales	1619	1253	818	**3690**

Tabla 2. Matriz de Confusión sin Stemming.

los resultados gráficamente. Como resultado porcentual generado por el clasificador se encontró que de los 2050 *tweets* de ciencia, se clasificaron correctamente 55 %, equivalente a 1133 *tweets*. Para la clase de tecnología se obtuvo un resultado positivo del 58 % que corresponde a 364 sobre un total de 627 *tweets*.

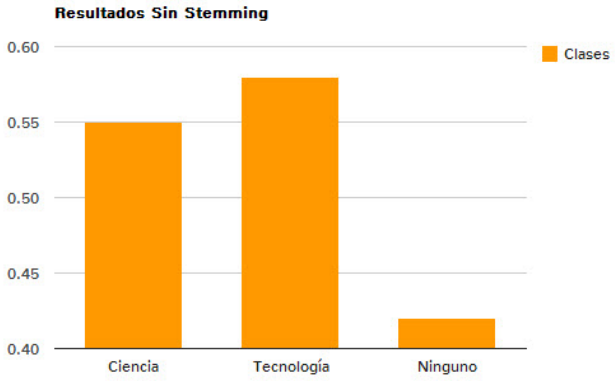

Figura 4. Resultados Sin Utilizar Stemming.

Finalmente en los *tweets* sin clasificación se obtuvo una precisión del 42 %. Esto quiere decir que de 1013 *tweets* únicamente se clasificaron exitosamente 427 *tweets*. La Tabla 3 puntualiza el resultado según la precisión y el *recall*.

Precisión	Ciencia - 55 %	Tecnología - 58 %	Ninguno - 42 %
Recall	Ciencia - 70 %	Tecnología - 29 %	Ninguno - 52 %

Tabla 3. Precisión y Recall, Sin *Stemming*.

Prueba Utilizando *Stemming*. Utilizando las técnicas de *stemming*, limpieza y tokenización, la matriz de confusión dio como resultado una precisión del 84 % junto con la generación de los siguientes resultados (Tabla 4):

	Ciencia	**Tecnología**	**Ninguno**	**Totales**
Ciencia	1745	170	135	2050
Tecnología	32	583	12	627
Ninguno	145	82	786	1013
Totales	1922	835	933	**3690**

Tabla 4. Matriz Utilizando *Stemming*.

En la Figura 5 podemos observar los resultados generados al utilizar *Stemming*. El clasificador Naïve Bayes indicó que de los 2050 *tweets* catalogados

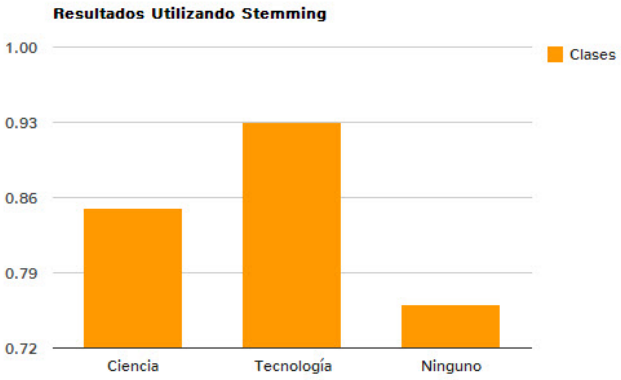

Figura 5. Resultados Utilizando *Stemming*.

como ciencia, se clasificaron 1745 correctamente dando como resultado porcentual 85 % para la clase tecnología un 93 % teniendo en cuenta que de los 627 se han clasificado 853 *tweets* correctamente. En último lugar los *tweets* fichados como ambiguos 786 han sido clasificados correctamente dando como porcentaje un 77 % (ver Tabla 5).

Precisión	Ciencia - 85 %	Tecnología - 93 %	Ninguno - 77 %
Recall	Ciencia - 91 %	Tecnología - 70 %	Ninguno - 84 %

Tabla 5. Precisión y Recall, utilizando *Stemming*.

5.1. Interpretación de los Resultados

En la primera prueba el clasificador obtiene una precisión del 52.14 % contra la segunda prueba que dio un resultado de 84 % pre-procesando los datos con *Stemming*. Sin embargo, la precisión en el campo de la Recuperación de Información difiere de las definiciones de exactitud y precisión en otras áreas.

Por tanto para determinar la exactitud del clasificador en ambos conjuntos, se utiliza la medida F1. A continuación se describe en qué consiste la precisión y la recuperación.

- Precisión: es el número de resultados correctos dividido por el número de todos los resultados devueltos.
- Recuperación: es el número de resultados correctos dividido por el número de resultados que deberían haber sido devueltos.

La puntuación de F1 se puede interpretar como una media ponderada de la precisión y la recuperación, donde una puntuación F1 alcanza su mayor valor en 1 y peor puntuación en 0. La Tabla 6 muestra las fórmulas que se han utilizado para el cálculo de los resultados. Las Tablas 7 y 8, muestran la clasificación del

Accuracy	(true positives + true negatives) / (total examples)
Precision	(true positives) / (true positives + false positives)
Recall	(true positives) / (true positives + false negatives)
F1 score	(2 * precision * recall) / (precision + recall)

Tabla 6. Fórmulas.

número total de emphtweets verdaderamente bien clasificados ya que la búsqueda del texto en los documentos seleccionados proporcionan una única medición para el sistema.

Ciencia 88 %
Tecnología 80 %
Ninguno 81 %
F1 83 %

Tabla 7. Resultados utilizando *Stemming*.

Ciencia 61 %
Tecnología 38 %
Ninguno 46 %
F1 48 %

Tabla 8. Resultados sin *Stemming*.

6. Conclusiones

De los resultados obtenidos podemos concluir que la utilización del *Stemming* como técnica de preprocesamiento de los datos mejora considerablemente el desempeño del clasificador, el Análisis léxico para analizar *tweets* resulta de gran utilidad para determinar a que categoría pertenece la información publicada por los usuarios.

El clasificador Bayesiano presenta un excelente rendimiento, debido principalmente a su robustez frente al ruido. Es un algoritmo muy práctico y sencillo de utilizar, especialmente porque hace múltiples suposiciones para simplificar el problema. Los cálculos probabilísticos realizados fueron acertados, la probabilidad de cada una de las palabras P(W) fue condicionalmente independiente dado el valor de las clases (Ciencia, Tecnología o Ninguno), así mismo con las entradas sobre nuevas clases también se reflejó un número de aciertos aceptable. El clasificador también demostró ser muy potente en cuanto al volumen de los datos que pueden procesar en tan poco tiempo. Resultó muy ventajoso clasificar cada uno de los *tweets* por su contenido en lugar de hacerlo por sus *hashtags*, ya que no todos los *tweets* tienen *hashtags*, y en muchas ocasiones los mismos usuarios les asignan una categoría que no es del todo cierta, dado eso y a lo dinámico que puede resultar la utilización de un *hashtag* (es difícil predecir cuál será su estructura). Como resultado se determinó que era más útil para la investigación trabajar con el contenido del *tweet*.

La enorme cantidad de información y su crecimiento exponencial en *Twitter* representa un reto para el Análisis y la recuperación de información, de la investigación realizada, se deriva el desarrollo de una aplicación capaz de utilizar diferentes técnicas de minería de texto, la aplicación es un primer acercamiento al Análisis léxico en *Twitter*, la investigación puede despertar un gran interés en profundizar en esta temática.

Bibliografía

[1] Ronald S Burt, M K, *Social Network Analysis Foundations and Frontiers on Advantage*. University of Chicago, University College London, University of Cambridge, 2013.

[2] Semiocast, 2012. [Internet; descargado 20-abril-2013: http://semiocast.com].

[3] J. Z. Maggie Johnson, *Lexical Analysis*. Stanford University, California, 2008.

[4] K. D. Benavides, *Procesamiento de Texto*. 2008.

[5] linqtotwitter, "2006-2013 microsoft," 2013. [Internet; descargado 20-abril-2013: http://linqtotwitter.codeplex.com].

[6] OAuth, 2007. [Internet; descargado 20-abril-2013: http://oauth.net].

[7] M. Bramer, "Introduction to classification: Naïve bayes and nearest neighbour," in *Principles of Data Mining, Undergraduate Topics in Computer Science*, pp. 21–37, Springer-Verlag, London, 2013.

[8] V. K.-H. XindongWu, "Top 10 algorithms in data mining," in *IEEE International Conference on Data Mining*, p. 37, Verlag London Limited, Springer, 2007.

[9] J. Kagström, *uClassify Classification Server Manual*. Spring, SMid Sweden: Mid Sweden University, 2005.

[10] M. Porter, *Strategy and the Internet*. Harvard Business Review, 2001.

[11] García, C., Alonso J. L, "Clasificación automática de documentos. un caso práctico," tech. rep., Instituto Universitario de estudios en Ciencia y Tecnología. España: Universidad de Salamanca, 2012.

[12] G. A. Miller, 2012. [Internet; descargado 16-mayo-2013: http://wordnet.princeton.edu].