Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Recuperación automática de fotografías y anécdotas en Facebook

Tesis

Laura Mariana Aguirre Moro

Ciudad Universitaria, CDMX 2016

Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Recuperación automática de fotografías y anécdotas en Facebook

Tesis

Que para obtener el título de:

Licenciado en Informática

Presenta:

Laura Mariana Aguirre Moro

Asesor:

Dr. Carlos Francisco Méndez Cruz

Ciudad Universitaria, CDMX. 2016

Contenido

Agradecimientos 1

Resumen 2

1. Introducción 3

1.1. Problema de investigación 3

1.2. Justificación del tema 4

1.3. Preguntas de investigación 5

1.4. Hipótesis de investigación 5

1.5. Objetivos del trabajo 5

1.6. Metodología 6

1.6.1. Recuperación automática de anécdotas mediante aprendizaje automático 6

1.6.2. Recuperación automática de fotografías mediante técnicas de recuperación de información 7

1.7. Alcance y limitaciones 7

1.8. Aportaciones 8

1.9. Estructura capitular 8

2. Marco teórico 10

2.1. Minería de textos 10

2.2. Clasificación automática 14

2.2.1. Aprendizaje automático 14

2.2.2. Definición 15

2.2.3. Algoritmos 15

2.2.4. Etapas de entrenamiento y evaluación 19

2.2.5. Reducción de dimensionalidad 21

2.3. Recuperación de información 22

2.3.1. Definición 23
2.3.2. Espacio vectorial 24
2.3.3. Medida TF-IDF 25
2.3.4. Evaluación 26
3. Recuperación automática de anécdotas 27
3.1. Recopilación del corpus de comentarios 27
3.2. Etiquetado lingüístico del corpus 30
3.3. Clasificación manual 33
3.4. Entrenamiento 36
3.5. Preprocesamiento 37
3.5.1. Experimentos 37
3.5.1.1. Baseline 42
3.6. Evaluación y resultados 43
3.6.1. Evaluación 43
3.6.2. Resultados 47
3.6.3. Discusión 52
4. Recuperación automática de fotografías 59
4.1. Recopilación del corpus 59
4.2. Preprocesamiento 62
4.3. Obtención de descriptores de fotografías 62
4.4. Evaluación y resultados 65
4.4.1. Evaluación 65
4.4.2. Resultados 69
5. Conclusiones 75
5.1. Resumen de experimentos 75
5.2. Revisión de preguntas, hipótesis y objetivos de investigación 76

5.3. Debilidades y fortalezas de los métodos propuestos.....80

5.4. Trabajo futuro.....80

5.5. Comentarios finales.....80

6. Anexos.....82

7. Referencias.....92

Índice de tablas

Tabla 3-1 Etiquetas para verbos 32

Tabla 3-2 Etiqueta para verbo en pasado canté 32

Tabla 3-3 Distribución de comentarios en corpus de entrenamiento y evaluación 36

Tabla 3-4 Representación de matriz para clasificación automática 39

Tabla 3-5 Parámetros de entrada del programa entrenarEvaluar.py 40

Tabla 3-6 Expresiones regulares para identificar verbos con etiquetas EAGLES 41

Tabla 3-7 Estructura de una matriz de confusión 44

Tabla 3-8 Ejemplo de una matriz de confusión 45

Tabla 3-9 Experimentos con categoría gramatical 48

Tabla 3-10 Vector binario con algoritmo SVM 49

Tabla 3-11 Vector conteo con algoritmo SVM 49

Tabla 3-12 Vector TF-IDF con algoritmo SVM 50

Tabla 3-13 Experimentos con algoritmo Naive Bayes 51

Tabla 3-14 Modos y tiempos verbales 51

Tabla 3-15 Baselines 52

Tabla 3-16 Mejores clasificadores por *accuracy* 53

Tabla 3-17 Matriz de confusión mejor clasificador por *accuracy* (ID 10) 54

Tabla 3-18 Matriz de confusión mejor clasificador por *accuracy* (ID 43) 54

Tabla 3-19 Mejor clasificador por *precision* 55

Tabla 3-20 Matriz de confusión mejor clasificador por *precision* 56

Tabla 3-21 Mejor clasificador por *recall* 56

Tabla 3-22 Matriz de confusión mejor clasificador por *recall* 57

Tabla 3-23 Mejor clasificador por F-score 57

Tabla 3-24 Matriz de confusión mejor clasificador por F-score 58

Tabla 4-1 Ejemplo de descriptores de una imagen 64

Tabla 4-2 Perfiles para evaluación de palabras clave generadas con TODOS los comentarios.....67

Tabla 4-3 Perfiles para evaluación de palabras clave generadas solo con los comentarios NO anécdotas.....67

Tabla 4-4 Resultados para ambos métodos.....69

Tabla 4-5 Resultados por tipo de búsqueda.....70

Tabla 4-6 Resultados por edad.....71

Tabla 4-7 Resultados por nivel de estudio.....72

Tabla 4-8 Resultados por tipo de búsqueda y edad.....72

Tabla 4-9 Resultados por tipo de búsqueda y nivel de estudio.....73

Índice de figuras

Figura 2-1 Clasificación de documentos 12

Figura 2-2 Extracción de información 14

Figura 2-3 Separación del hiperplano en dos clases 16

Figura 2-4 Posibles hiperplanos de separación en dos clases 17

Figura 2-5 Vectores de soporte que definen el hiperplano 17

Figura 2-6 Representación de la construcción de un modelo clasificador 20

Figura 2-7 Recuperación de información 23

Figura 2-8 Etapas de la recuperación de información 23

Figura 2-9 Matriz de términos de documento 24

Figura 3-1 Salida de una consulta en formato Json 29

Figura 4-1 Salida de consulta a la tabla photo 60

Figura 4-2 Salida de la consulta a la tabla comment 61

Figura 4-3 Fotografía del álbum La ciudad Universitaria 63

Figura 4-4 Interfaz para el buscador de imágenes 66

Figura 4-5 Interfaz de evaluación de búsqueda 68

Figura 5-1 Método para la detección automática de anécdotas en los comentarios de la página del Facebook La ciudad de México en el tiempo 78

Figura 5-2 método para la recuperación automática de fotografías del sitio de Facebook La ciudad de México en el tiempo 79

Agradecimientos

A Dios por la fortaleza que me da cada día para seguir adelante.

A mi familia por la educación y apoyo que me brindaron, son lo más especial que tengo, gracias por confiar en mí. A mi mamá por ser un ejemplo de ayudar a otros, a mi papá por el esfuerzo que depositó para crecer como familia, a mi hermana por ser un ejemplo de fortaleza y a mi sobrino por ser la luz que me motiva a seguir cada día.

A mis amigos y compañeros que me acompañaron durante la carrera, son una parte muy importante de mi vida, aprendí muchas cosas de ellos, gracias por confiar en mí, por brindarme su amistad, por las pláticas, las risas y por ser un gran apoyo para enfrentar mis miedos.

A mi asesor el Dr. Carlos Francisco Méndez Cruz por compartir sus conocimientos durante la carrera, por todo su apoyo durante mi estancia como becaria en el Grupo de Ingeniería Lingüística, por su tiempo, paciencia y dedicación en el desarrollo de esta tesis y por ser un gran ejemplo como persona y profesional. Gracias.

A la Universidad Nacional Autónoma de México por el gran esfuerzo que realiza día a día para formar profesionistas útiles a la sociedad, es un gran orgullo pertenecer a esta casa de estudios.

A la Facultad de Contaduría y Administración y a los profesores de la misma por su esfuerzo y dedicación en cada una de sus clases que me permitieron adquirir nuevos conocimientos.

Al Grupo de Ingeniería Lingüística(GIL) donde desarrollé esta tesis. Finalmente, al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca que me otorgó a través de los proyectos *Caracterización de huellas textuales para análisis forense* (Clave de proyecto: 215179) y *Sistemas para procesar automáticamente documentación relacionada con la biblioteca de arte mexicano* (Clave de proyecto: 221247).

Resumen

Esta tesis tiene como objetivo desarrollar un método para la recuperación automática de anécdotas y fotografías de la página de Facebook *La ciudad de México en el tiempo*. En dicha página se publican imágenes históricas de la ciudad de México y algunos de los comentarios que se realizan en ellas pueden ser considerados anécdotas.

El objetivo de recuperar fotografías y anécdotas es crear una memoria colectiva que pueda ser consultada por diferentes personas. Actualmente este proceso se realiza manualmente y uno de los principales problemas es que hay una gran cantidad fotografías publicadas y es muy difícil la búsqueda de fotografías y anécdotas.

Los métodos propuestos están basados en diferentes técnicas, las anécdotas se recuperan a través de un proceso de clasificación automática y las fotografías se recuperan a través de palabras clave con técnicas de recuperación de información. Se llevaron a cabo diferentes experimentos con el fin de encontrar el mejor método para la recuperación automática de fotografías y anécdotas.

1. Introducción

Actualmente vivimos rodeados de información y cada día se genera más utilizando Internet como principal recurso para compartirla. Esta red mundial ha permitido que tengamos acceso a dicha información de una manera más fácil. Mucha de esa información se comparte específicamente en las redes sociales.

Según la Asociación mexicana de Internet (AMIPCI, 2015)[1], en México el acceso a redes sociales como Facebook y Twitter es la principal actividad online. Por ello, muchas empresas se han enfocado y han realizado estudios para obtener información de redes sociales, ya que cada día se generan opiniones, comentarios, ideas, etc. Además, la facilidad con la que se pueden compartir nuevos temas a través de las redes sociales es sobresaliente.

1.1. Problema de investigación

Un ejemplo de lo dicho anteriormente es la página de Facebook *La ciudad de México en el tiempo*. Esta es un espacio para compartir fotografías, anécdotas e información acerca de la historia de la ciudad de México con fines educativos y de difusión cultural. Fue creada desde el año 2011 y actualmente cuenta con 441,212 *likes*, los cuales aumentan cada día al igual que la información que se recopila en esta página. Este sitio ha llegado a tener un valor histórico por las fotografías que se publican. Además, la gente cuenta anécdotas o recuerdos de cierto tiempo y lugar, los cuales tienen un valor cultural e histórico. En el mes de abril de 2014 existían 256,059 comentarios. Lo mismo sucede con las fotografías, en esa misma fecha había 5,217 imágenes publicadas y estas cifras aumentan cada día.

Estos comentarios podrían ser fuente de estudio para diversas disciplinas como la historia, la antropología, la lingüística y en general los estudios culturales. De hecho, los administradores del sitio han recibido diversas invitaciones para presentar sus fotografías en libros, publicaciones y exposiciones. Algunos ejemplos son *Ciudad Sueño y Memoria* (Pérez Gay, de Mauleón, & Villasana Suaveza, 2013) *Revolución e Imagen*, Revista Alquimia - Sistema Nacional de Fototecas SINAFO 2010, la exposición *60 años de actividades académicas en la Ciudad Universitaria*, Museo Universitario de Ciencias y Artes (MUCA), entre otras.

[1] https://www.amipci.org.mx/images/AMIPCI_HABITOS_DEL_INTERNAUTA_MEXICANO_2015.pdf

Por otra parte, la riqueza de los comentarios ha llevado a querer recuperar colecciones de anécdotas y recuerdos para crear un tipo de memoria colectiva extraída de los contenidos de la página. Sin embargo, la gran cantidad de comentarios existentes en el sitio provoca que la lectura y análisis de cada uno sea prácticamente imposible o al menos lleve demasiado tiempo.

Para la selección de fotografías sucede algo similar. La gran cantidad de material gráfico del sitio hace difícil encontrar imágenes específicas. Dado que no existe en Facebook un buscador que recupere fotografías de una página, su búsqueda se hace consultando cada uno de los álbumes de la página. Un método para buscar fotografías ayudaría a que su selección y recuperación sea de forma más rápida.

Por lo tanto, es necesario el desarrollo de tecnologías que ayuden a recuperar automáticamente información en redes sociales. Así esta tesis busca resolver este problema mediante la propuesta de un método para recuperar automáticamente anécdotas y fotografías (imágenes) en Facebook.

1.2.Justificación del tema

El recurso más importante que la raza humana posee es el conocimiento y este conocimiento sigue existiendo y creándose de forma continua en forma de documentos, libros, artículos, etc. (Gelbukh & Sidorov, 2010). Este conocimiento ahora se almacena también en formato electrónico, o sea, digital. Por esta razón es necesario el uso de tecnologías que permitan el procesamiento de la información contenida en ese formato.

Por otra parte, la Informática juega un papel muy importante en el manejo, procesamiento y almacenamiento de información teniendo como objetivo la automatización de diferentes tareas. Tradicionalmente, la Informática se ha dedicado al procesamiento de información estructurada que proviene de bases de datos. Sin embargo, hoy en día también tenemos acceso a enormes cantidades de información no estructurada, como texto, audio y video. Por tanto, la Informática tiene que tomar en cuenta el procesamiento y almacenamiento de este tipo de información.

La página de *La ciudad de México en el tiempo* tiene grandes cantidades de información que pueden ser procesadas para lograr la automatización de tareas, en este caso, la recuperación de imágenes y detección de recuerdos o anécdotas con el fin de ser consultados para diferentes estudios o investigaciones. Actualmente no conocemos ningún método automatizado para la

detección de recuerdos o anécdotas en comentarios de Facebook o para la recuperación automática de imágenes mediante palabras clave obtenidas de los comentarios de Facebook.

Así, mediante la colaboración de la informática, la lingüística y el procesamiento del lenguaje natural se puede desarrollar un método para realizar estas tareas automáticamente.

1.3. Preguntas de investigación

Las preguntas de investigación de las cuales parte esta tesis son las siguientes:

- ¿Será posible la detección automática de recuerdos y anécdotas a partir de los comentarios generados en una página de Facebook?
- ¿Será posible la recuperación automática de imágenes (fotografías) publicadas en Facebook?

1.4. Hipótesis de investigación

En esta tesis se ha planteado la siguiente hipótesis de investigación:

1. Sí es posible la detección automática de recuerdos y anécdotas a partir de los comentarios generados en una página de Facebook.
2. Sí es posible la recuperación automática de imágenes (fotografías) publicadas en Facebook a través de la generación automática de descriptores.

1.5.Objetivos del trabajo

Dados el problema, las preguntas y las hipótesis de investigación, esta tesis tiene como objetivos principales:

1. Desarrollar un método para la detección automática de anécdotas en los comentarios de la página de Facebook *La ciudad de México en el tiempo.*
2. Desarrollar un método para la recuperación automática de fotografías (imágenes) del sitio de Facebook *La ciudad de México en el tiempo* mediante la generación automática de descriptores.

Además, como objetivos secundarios se pretende:

- Investigar sobre técnicas de minería de datos, específicamente sobre métodos de clasificación dentro del aprendizaje automático.
- Desarrollar modelos de clasificación para detección de recuerdos o anécdotas.
- Investigar sobre generación automática de descriptores a partir de comentarios de redes sociales.

1.6.Metodología

Ya que esta tesis tiene dos principales objetivos, se utilizarán dos metodologías distintas para cada uno de ellos. Para la detección automática de recuerdos y anécdotas se utilizarán técnicas de aprendizaje automático y para la recuperación automática de imágenes se utilizan técnicas de recuperación de información. A continuación, se describen las metodologías a seguir para cumplir cada uno de estos objetivos.

1.6.1. Recuperación automática de anécdotas mediante aprendizaje automático

1. Recopilación del corpus (datos)

El corpus (datos) que se utilizará está conformado por comentarios de la página de Facebook *La ciudad de México en el tiempo.* Entonces, el primer paso consiste en la recopilación de los comentarios. Para ello utilizaremos *Graph API* y el *Facebook Query Language (FQL)*, que son herramientas que provee Facebook a los desarrolladores.

2. Preprocesamiento

Los comentarios obtenidos serán preprocesados de modo que se obtenga cada comentario en un documento separado y en texto plano.

3. Clasificación manual

Una vez que tenemos un comentario en cada documento, el siguiente paso consiste en asignar una etiqueta a cada documento dependiendo de si es anécdota o no es anécdota. Este paso permite brindar a la máquina ejemplos de datos clasificados (experiencia) de la cual se puede obtener un modelo clasificador mediante aprendizaje automático. En este punto también se divide el corpus. Una parte es utilizada para la fase entrenamiento y otra para la fase de evaluación.

4. Entrenamiento (realizar experimentos para crear modelos clasificadores)

Con el corpus de entrenamiento clasificado se realizan experimentos tomando en cuenta diferentes características de los comentarios y de esta forma se crean diferentes modelos clasificadores.

5. Evaluación

Para llegar a una conclusión sobre el mejor modelo clasificador, se utilizarán los modelos clasificadores obtenidos de la fase de entrenamiento serán aplicados en el corpus de evaluación. Se tomará en cuenta la medida de evaluación llamada *Accuracy* para poder comparar los diferentes modelos clasificadores y elegir cuál se utilizará para lograr la detección automática de anécdotas.

1.6.2. Recuperación automática de fotografías mediante técnicas de recuperación de información

1. Recopilación del corpus (datos)

El corpus (datos) que se utilizará está conformado por comentarios de la página de Facebook *La ciudad de México en el tiempo.* Entonces, como ya se mencionó, el primer paso consiste en la recopilación de los mismos con *Graph API* y el *Facebook Query Language (FQL)*.

2. Preprocesamiento

Los comentarios obtenidos serán preprocesados de modo que se obtengan todos los comentarios de una imagen en un solo documento y en texto plano, de esta forma se obtiene un corpus de comentarios de imágenes.

3. Obtener los descriptores de las imágenes

Esto se logrará a través de dos procesos. El primero consiste en calcular medidas de TF-IDF de las palabras de cada documento. El segundo consiste en obtener las palabras más relevantes en un documento a partir de esa medida. Estas palabras relevantes serán tomadas como descriptores para la fotografía.

4. Evaluación

Se realizará una evaluación manual para revisar si las palabras obtenidas describen la fotografía con la que se relacionan.

1.7. Alcance y limitaciones

En este trabajo de investigación solo se ocupará aprendizaje automático como metodología para la detección de anécdotas, dejando para futuras investigaciones otro tipo metodologías. Al respecto, ya que es la primera vez que se busca un método para recuperar anécdotas en

comentarios de redes sociales, solo se utilizarán dos algoritmos de aprendizaje automático (SVM y MultinomialNB) dejando para trabajo futuro comparar con otros algoritmos.

Solo se procesarán comentarios de la página de Facebook *La ciudad de México en el tiempo*, no se tomarán en cuenta otras páginas. Finalmente, solo se usarán comentarios escritos en lengua española, aunque los métodos que utilizaremos son relativamente independientes del idioma.

1.8.Aportaciones

Las aportaciones inmediatas serían para los administradores de la página de Facebook *La Ciudad de México en el tiempo* ya que les permitiría contar con un método automático para recuperar fotografías a partir de palabras de búsqueda. Además, tendrían a su disposición un método para recuperar anécdotas que puedan ser incluidas en memorias colectivas y libros sobre la página.

Por otra parte, esta tesis aportaría a la investigación en Informática, especialmente en procesamiento automático de información, una base que sirva para el desarrollo de métodos y estudios para la recuperación de información textual (anécdotas) y visual (fotografías) de una red social. En otras palabras, esta tesis aporta al desarrollo de métodos automáticos de análisis de redes sociales.

1.9.Estructura capitular

Esta tesis se estructura de la siguiente manera. El capítulo presente es la introducción, donde se presenta el problema, objetivos, metodología, aportaciones, alcances y limitaciones de la tesis.

En el segundo capítulo que es el marco teórico se abordan temas relacionados a la investigación como minería de textos, clasificación automática, recuperación de información y otros con el fin de conocer los diferentes métodos que se utilizan para realizar tareas de análisis de textos.

El tercer capítulo se describe todo el proceso que se realizó para la generación automática de anécdotas, desde la recopilación de comentarios de la página de Facebook *La ciudad de México en el tiempo,* la creación, etiquetado y distribución del corpus. Describe también los experimentos, es decir, los algoritmos y características que se utilizaron para encontrar el mejor modelo clasificador de anécdotas y recuerdos.

En el capítulo cuatro se presenta el proceso para la obtención de descriptores de imágenes, desde la recopilación de información, el método propuesto para conocer que palabras describen mejor a una imagen y finalmente se describe la forma de evaluar a estos descriptores.

Finalmente se presentan las conclusiones de la investigación, el resumen de los experimentos, revisión de preguntas e hipótesis planteadas al inicio de la tesis, debilidades, fortalezas y trabajo a futuro.

2. Marco teórico

2.1.Minería de textos

Para introducirnos en el tema de minería de textos es importante conocer un poco a cerca de la minería de datos ya que son dos tareas que se relacionan de cierta forma. Para Elmasri (2007) la minería de datos tiene relación con la extracción de nueva información en términos de patrones o reglas a partir de grandes cantidades de datos. Para lograr esta tarea de extraer información a partir de datos la minería de datos se auxilia de diferentes técnicas tomadas de áreas como la estadística, la inteligencia artificial y sistemas de bases de datos.

Los principales objetivos de realizar minería de datos son:

1.- Objetivos predictivos: Es decir que logra, mediante el uso de una parte de variables, predecir una o más de otras variables. Como ejemplo está la clasificación y detección de anomalías o valores atípicos (Gorunescu F. , 2011).

2.- Objetivos descriptivos: Se refiere a la identificación de patrones que describen los datos y que pueden ser fácilmente comprendidos por el usuario. Por ejemplo, el agrupamiento, el descubrimiento de reglas de asociación y el descubrimiento de patrones. (Gorunescu F. , 2011)

Actualmente se utiliza la minería de datos para aplicaciones en diferentes áreas como la economía, medicina, investigación, y en general diferentes empresas utilizan minería de datos para encontrar patrones o información importante que ayude a mejorar las ventas y procesos de la organización.

Por otro lado, para entender la diferencia entre minería de datos y minería de textos es importante entender que actualmente se procesan grandes cantidades de datos estructurados y no estructurados. Hablamos de datos estructurados cuando estos están contenidos en una estructura como una base de datos y, por el contrario, los no estructurados hacen referencia a los formatos de texto como correo electrónico o páginas web que no tiene una estructura para poder ser procesados por sistemas de información.

Los métodos de minería de datos procesan información estructurada numérica, mientras que el texto se describe a menudo como información no estructurada. El texto y los documentos pueden transformarse en valores tales como la presencia o ausencia de palabras y los mismos métodos que han demostrado ser exitosos en la minería de datos se pueden aplicar al texto. (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010).

Para Gelbukh & Sidorov (2010) la minería de texto consiste en descubrir conocimiento que no está literalmente escrito en grandes cantidades de texto incluyendo la búsqueda de tendencias, promedios, desviaciones, dependencias, etc.

Tomando en cuenta los conceptos de minería de datos y minería de texto podemos concluir que la minería de datos trata de buscar patrones en los datos, del mismo modo, la minería de texto se trata de la búsqueda de patrones en el texto, es decir, es el proceso de análisis de texto para extraer información que es usada para un propósito en particular. (H. Witten, Frank, & A. Hall, 2011).

La minería de textos se auxilia de diferentes áreas como la inteligencia artificial y la lingüística que componen a la lingüística computacional, esta última tiene como objetivo la comprensión del lenguaje hablado o escrito a una representación estructurada o semiestructurada, de modo que pueda ser procesado.

En el proceso de minería de texto existen dos principales etapas: la etapa de preprocesamiento y la etapa de descubrimiento. En la primera etapa el texto es transformado a una representación estructurada o semiestructurada y la segunda tiene como objetivo descubrir patrones interesantes o nuevo conocimiento a partir de la representación estructurada. (Tan, 1999)

Además, la minería de texto es un campo multidisciplinario, que involucra la recuperación de información, análisis de texto, extracción de información, clustering (agrupamiento), categorización, visualización de base de datos, tecnología, aprendizaje automático y minería de datos. Según Weiss (2010) en las siguientes áreas se aplican diferentes métodos de minería de texto:

1. Clasificación de documentos

   En esta tarea se busca realizar una categorización del texto. Dada una muestra de experiencia pasada y respuestas correctas para cada ejemplo, el objetivo es encontrar nuevas respuestas para nuevos ejemplos. Esto se realiza transformando el texto en datos numéricos, generalmente se realiza en formato de tabla (matriz numérica) con las etiquetas (clases) a las que pertenece cada documento. Una vez que los datos se transforman en este formato, pueden ser aplicados los métodos de minería de datos. (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010).

En la Figura 2-1 podemos observar la representación de la tarea de clasificación de documentos, en donde los documentos son organizados en folders según la clase o categoría a la que pertenecen. A la entrada del proceso se encuentra un nuevo documento, el objetivo es colocar el nuevo documento en la carpeta correspondiente (clasificarlo o categorizarlo). En este ejemplo, se presenta una clasificación binaria en donde el documento puede estar o no en algún tema, por ejemplo, el nuevo documento tiene la opción de ser colocado o no en la carpeta finanzas.

En esta tesis se utilizó la clasificación automática para detectar anécdotas y no anécdotas. Dado lo anterior se presenta más adelante una sección completa dedicada a la clasificación automática.

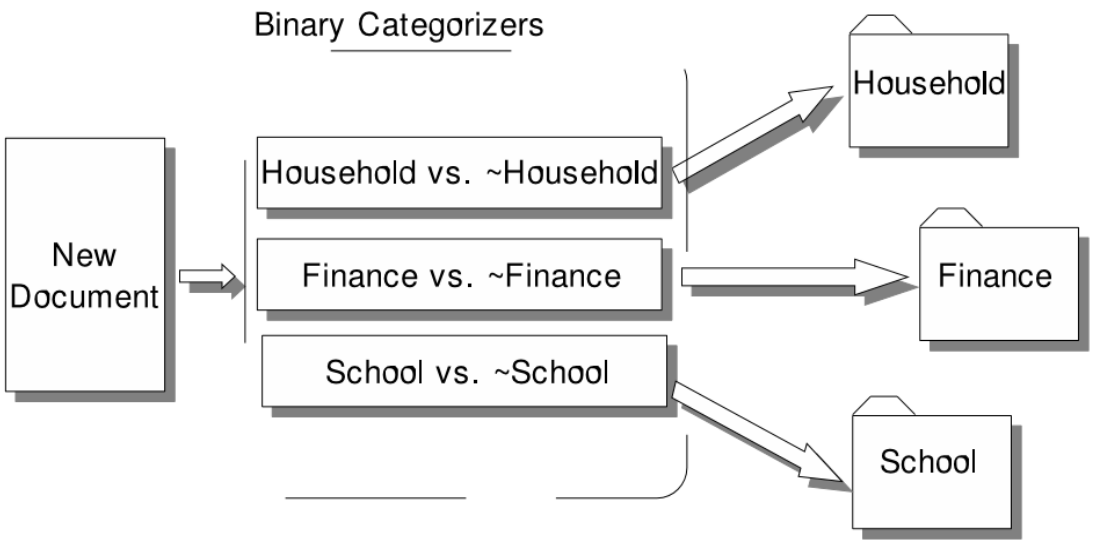


Figura 2-1 Clasificación de documentos

Tomada de Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F. (2010, pág. 7)

2. Recuperación de información

Para Gelbukh & Sidorov (2010) la recuperación de información trata de identificar cuales documentos (archivos de texto) son relevantes para una petición del usuario y cuáles no. En nuestro caso, la tarea consiste en obtener las fotografías relevantes, esto es, las que cumplen la petición del usuario en forma de palabras clave. Ya que esta tesis utiliza técnicas de recuperación de información, en la sección 2.3 se explicará con mayor detalle la tarea de Recuperación de información.

3. Clustering (agrupamiento)

   Como mencionamos anteriormente la clasificación de documentos se puede entender como el proceso para colocar nuevos documentos en carpetas correspondientes, dichas carpetas (clases) fueron establecidas previamente. A diferencia de la clasificación, en el *clustering* no existen carpetas preestablecidas por lo que busca establecer cuáles serán esas carpetas, es decir, crear nuevas carpetas para agrupar los documentos similares de acuerdo a su contenido. Por lo general esto se determina usando la frecuencia con que ciertos términos aparecen en los documentos. Para Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F. (2010), el proceso de agrupamiento es equivalente a la asignación de las etiquetas necesarias para la clasificación de textos.

4. Extracción de información

   Como mencionamos anteriormente, la primera etapa para realizar minería de texto es transformar el texto en datos estructurados o semiestructurados para que puedan ser procesados por computadoras. La extracción de información es un subproceso de la minería de texto que busca transformar el texto a los datos estructurados (generalmente a una base de datos) tal y como se utilizarían en la minería de datos (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010).

   Para Gelbukh & Sidorov (2010) la extracción de información es la transformación de algunas partes de los textos libres en un formato de base de datos, por ejemplo: fecha y lugar de un evento, precio de algún artículo, volumen de venta, etc.

   En la Figura 2-2 se representa el proceso de extracción de información, en donde podemos ver un documento de entrada donde se encuentran dos datos importantes que son el ingreso y la ganancia de una compañía, estos datos son transformados a un formato estructurado, en este caso, como tabla.

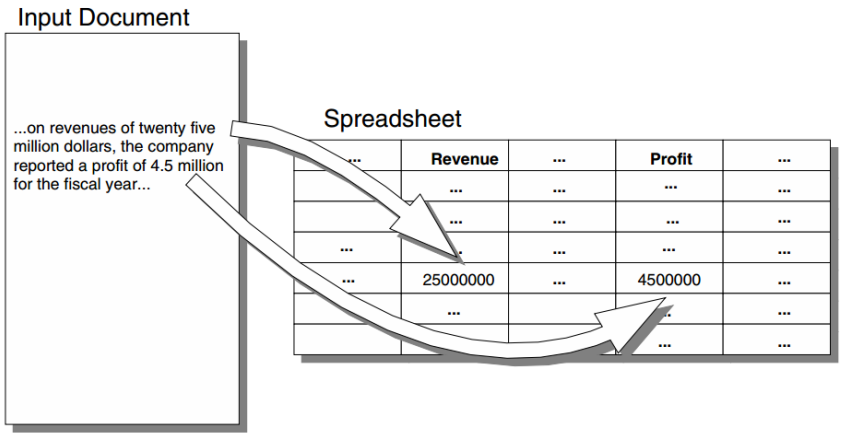


Figura 2-2 Extracción de información

Tomada de Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010 (2010, pág. 11)

Como ya se dijo, para realizar esta tesis nos basamos en dos principales áreas de la minería de texto que es la clasificación automática de documentos y la recuperación de información. En las siguientes secciones se describen estas áreas y los algoritmos utilizados.

2.2.Clasificación automática

Como ya mencionamos, una de las tareas de la minería de textos es la clasificación automática de documentos. Esta tarea involucra diferentes áreas, una de ellas es el aprendizaje automático. A continuación, se presentan los fundamentos de esta área. Posteriormente se presenta la definición, etapas y algoritmos para realizar clasificación automática.

2.2.1. Aprendizaje automático

Para hablar de clasificación automática es importante entender que es una tarea del aprendizaje automático o *Machine Learning*. Esta es una rama de la inteligencia artificial cuyo objetivo principal es hacer que las maquinas aprendan.

Entendemos como aprendizaje al proceso para obtener algo por medio del estudio, la experiencia o algo que se enseña. Un sistema de aprendizaje automático es un conjunto de algoritmos que resuelven problemas a través de la toma de decisiones que se basan en experiencias pasadas en las que se resolvió este mismo problema de manera correcta (Moreno, 1994).

El aprendizaje automático se puede aplicar en diferentes campos que van desde la política, biología, ciencias de la tierra, hasta las finanzas. Es una herramienta que se puede aplicar

a muchos problemas, cualquier campo que necesita interpretar y tomar decisiones con base a los datos o información puede beneficiarse de las técnicas de aprendizaje automático. (Harrington, 2012). Los métodos de aprendizaje automático se clasifican en supervisados y no supervisados considerando el tipo de estrategia que utilizan y el tipo de entrada que reciben los sistemas de aprendizaje.

Los métodos de aprendizaje supervisado se caracterizan porque los ejemplos proporcionados como entrada son necesarios para cumplir las metas de aprendizaje. En este tipo de aprendizaje se dan ejemplos y se especifica de qué conceptos son (Moreno, 1994). Un clásico ejemplo del aprendizaje supervisado es la clasificación.

El otro método de aprendizaje es el no supervisado en el cual se buscan automáticamente estructuras o grupos en el conjunto de datos de entrada (Gorunescu F. , 2011). Son diseñados para encontrar nuevo conocimiento mediante el descubrimiento de regularidades de los datos de entrada (Moreno, 1994). Un ejemplo de aprendizaje no supervisado en el *clustering* o agrupamiento.

2.2.2. Definición

Tomando en cuenta la categorización del aprendizaje automático, la clasificación automática es un método supervisado. Un sistema de clasificación automática recibe ejemplos que toma como base para poder clasificar nuevas entradas.

Según Gorunescu (2011), la clasificación automática se define como el proceso de colocación de un objeto en un conjunto especifico de categorías (clases) con base en las características (atributos) del objeto. También tenemos la definición que no da Weiss (2010) que dice que la clasificación consiste en encontrar clases correctas para un nuevo conjunto de datos (conjunto de datos de evaluación) a partir de experiencias con datos en donde las clases están correctamente asignadas (conjunto de datos de entrenamiento).

2.2.3. Algoritmos

Existen diferentes algoritmos en las tareas de aprendizaje automático supervisado que se utilizan para tareas de clasificación. Para esta tesis se utilizarán el algoritmo *Support Vector Machine* (SVM) y Naïve Bayes. A continuación, se presenta la explicación de estos dos algoritmos.

2.2.3.1. Support Vector Machine

Las Máquinas de Vector Soporte o SVM (Support Vector Machine), son una técnica de clasificación automática que es utilizada en diferentes problemas como multiclasificación, agrupamiento y regresión. En muchas aplicaciones, las SVM han logrado tener gran desempeño, incluso más que las máquinas de aprendizaje tradicional como las redes neuronales por lo que han sido introducidas como herramientas poderosas para resolver problemas de clasificación (Betancourt, 2005).

Las Maquinas de Vector Soporte toman datos de entrada que serán clasificados en una de dos clases establecidas. Utilizan un conjunto de datos de entrenamiento que están etiquetados con su clase correspondiente, el fin de utilizar un conjunto de datos de entrenamiento es tener una base para generar un modelo clasificador que será utilizado en un conjunto de datos de entrenamiento.

En un problema de clasificación que tiene dos clases con datos de entrenamiento separables se busca es una línea, también llamada hiperplano, que divida estas clases como se observa en un ejemplo en la Figura 2-3, sin embargo, existen otras posibles líneas que son capaces de agrupar estas clases como se puede observar en la Figura 2-4.

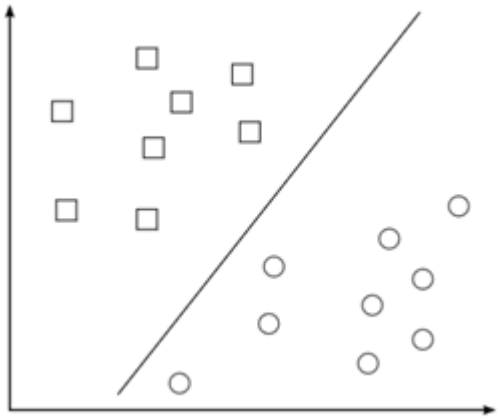


Figura 2-3 Separación del hiperplano en dos clases

Tomada de Suárez C. (2014, pág. 3)

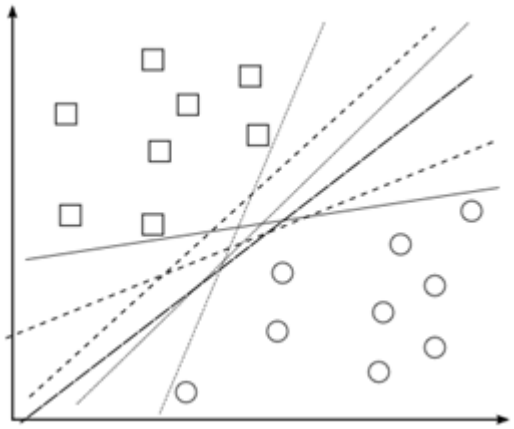


Figura 2-4 Posibles hiperplanos de separación en dos clases

Tomada de Suárez C. (2014, pág. 3)

En la Figura 2-5 cada elemento de una clase es representada como una figura(cuadrado y circulo). Para definir la línea (hiperplano) las SVM toman en cuenta la distancia que se encuentra lo más lejos posible de cualquier punto de los elementos de cada clase. La distancia entre el hiperplano y los puntos de datos más cercanos define el margen del clasificador, esto quiere decir que los puntos de datos más cercanos al hiperplano serán los que definan la posición de este. Estos puntos se representan en la Figura 2-5 y son llamados vectores de soporte.

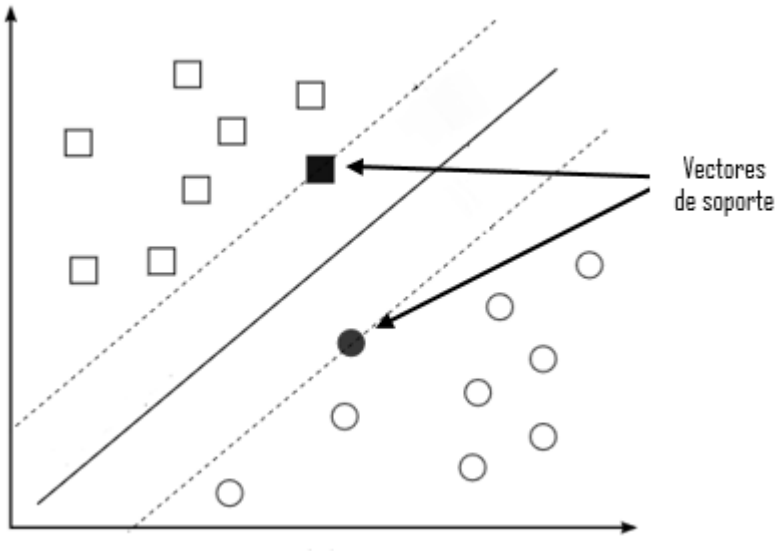


Figura 2-5 Vectores de soporte que definen el hiperplano

Tomada de Suárez C. (2014, pág. 4)

2.2.3.2. Naïve Bayes

El algoritmo Naïve Bayes es una técnica muy utilizada en la clasificación automática y puede llegar a ser muy rápido y exacto obteniendo buenos resultados de clasificación en la mayoría de los casos. Es un método supervisado por lo que necesita de un conjunto de datos de entrenamiento que contendrá los documentos ejemplo y de un conjunto de datos de evaluación para asignar clases a nuevos documentos. Naïve Bayes es un algoritmo probabilístico en el que cada documento es un conjunto de palabras, y cada ejemplo observado modifica la probabilidad de una hipótesis (asignar una clase) para nuevos documentos, únicamente aumenta o disminuye la probabilidad. Este algoritmo se base en el Teorema de Bayes conocido también como probabilidad condicional.

A continuación, se explican de forma general las fórmulas utilizadas en este algoritmo. Según Manning, Raghavan, & Schütze (2009), la probabilidad de un documento ***d*** de estar en clase ***c*** se calcula como sigue:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

donde $P(t_k|c)$ es la probabilidad condicional del término (palabra) t_k que ocurre en un documento de la clase *c***.** Para esta fórmula es importante mencionar que se utiliza una probabilidad de clase previamente establecida para un documento. $P(c)$ es la probabilidad previa (*a priori*) de que un documento pertenece la clase *c*. Si los términos de un documento no proporcionan una clara evidencia o prueba de una clase contra otra, elegimos la que tiene una mayor probabilidad *a priori*.

Asumamos que $< \mathrm{t1}, \mathrm{t2}, \ldots, \mathrm{t}n_d >$ son los términos de un documento *d* que forman parte del vocabulario que utilizamos para la clasificación, entonces n_d es el número de tales términos en *d*. Por ejemplo, $< \mathrm{t1}, \mathrm{t2}, \ldots, \mathrm{t}n_d >$ para el documento de una sola frase "*Beijing and Taipei join the WTO*" podría ser <Beijing, Taipei, join, WTO>, con n_d = 4, si descartando las *stop words*.

En la clasificación de texto, el objetivo es encontrar la mejor clase para cada documento. La mejor clase en es denominada la más probable o máximo a posteriori (MAP), que se representa como la clase Cmap y se obtiene con la siguiente fórmula:

$$Cmap = \underset{c \in C}{\arg\max} \left[\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} log\hat{P}(t_k|c) \right]$$

En esta fórmula se escribe $\hat{P}$ para *P* porque no conocemos los verdaderos valores de los parámetros $P(c)$ y $P(t_k|c)$, pero se estiman con el conjunto de entrenamiento.

En esta ecuación, cada parámetro condicional $log\hat{P}(t_k|c)$ es un peso que indica qué tan buen indicador es t_k para *c*. Del mismo modo, el $\hat{P}(c)$ es un peso que indica la frecuencia relativa de *c.*

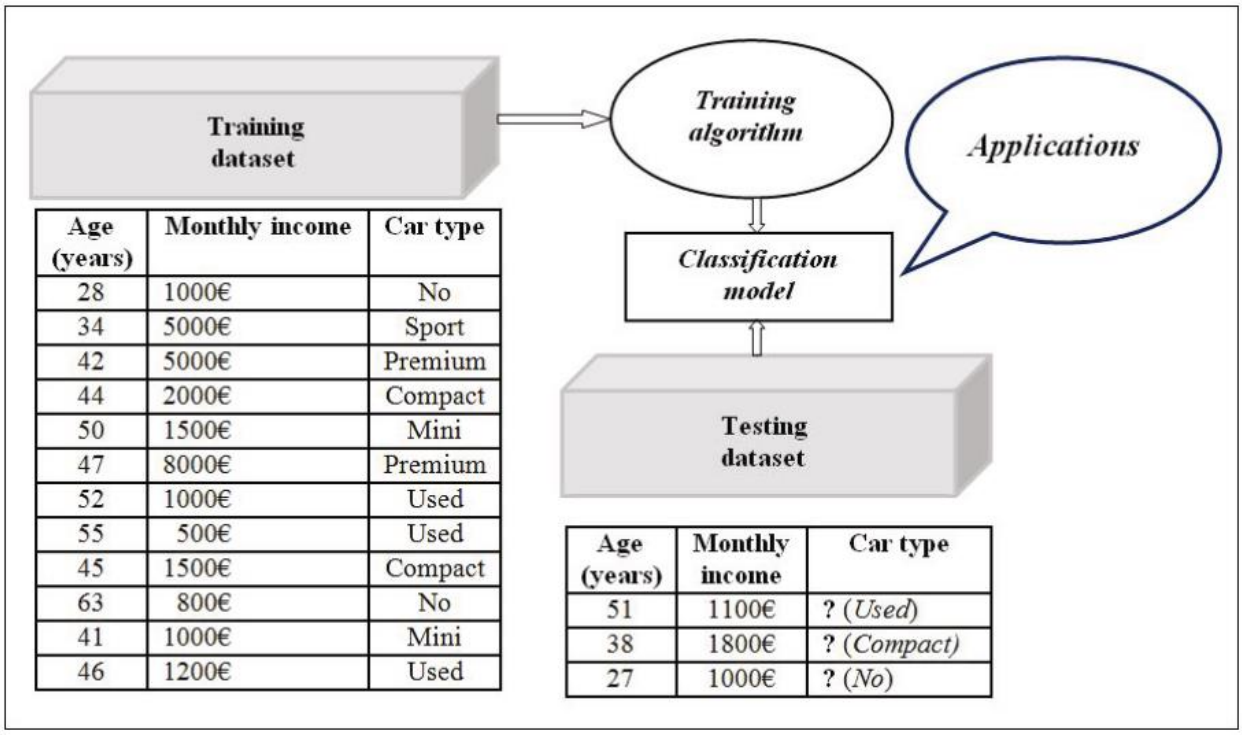
2.2.4. Etapas de entrenamiento y evaluación

Antes de describir las etapas del proceso de clasificación es importante conocer diferentes conceptos, según Gorunescu (2011) estos conceptos son la base del proceso de clasificación:

- Clase: Es una variable que representa la etiqueta que asigna un modelo clasificador a un objeto.
- Predictor: Son las variables independientes del modelo y son representados por las características o atributos de los datos que se clasifican. Por ejemplo, consumo de alcohol, estado civil, frecuencia de compra de algún producto, etc.
- Conjunto de datos de entrenamiento: Es el conjunto de datos que contiene valores para las clases y los predictores, es decir, de acuerdo a los atributos en el conjunto de datos de entrenamiento se establece a qué clase pertenece cada objeto; se utiliza para la creación del modelo clasificador. Algunos ejemplos de un conjunto de entrenamiento es información acerca de grupos de pacientes que se han tenido un ataque al corazón, de esta manera se puede crear un modelo clasificación con las características de estos pacientes. Otro ejemplo son grupos de clientes de un supermercado que compran un determinado producto.
- Conjunto de datos de prueba o evaluación: Es el conjunto que contiene nuevos datos que serán clasificados con el modelo creado a partir de los datos de entrenamiento con el fin de medir y evaluar el modelo.

Las etapas de la clasificación automática son entrenamiento y evaluación, en la Figura 2-6 se muestra el ejemplo del proceso de construcción de un modelo clasificador que propone

Gorunescu F. (2011), es un modelo clasificador para determinar el tipo de auto que podría comprar una persona según su edad e ingreso mensual, construyendo así un perfil de comprador.



Age (years)	Monthly income	Car type
28	1000€	No
34	5000€	Sport
42	5000€	Premium
44	2000€	Compact
50	1500€	Mini
47	8000€	Premium
52	1000€	Used
55	500€	Used
45	1500€	Compact
63	800€	No
41	1000€	Mini
46	1200€	Used

Age (years)	Monthly income	Car type
51	1100€	? (*Used*)
38	1800€	? (*Compact*)
27	1000€	? (*No*)

Figura 2-6 Representación de la construcción de un modelo clasificador

Tomada de Gorunescu F. (2011, pág. 17)

Para el proceso de clasificación automática se utilizan dos conjuntos de datos, uno de entrenamiento y otro de evaluación. La primera etapa del proceso de clasificación automática es el entrenamiento en donde se construye el modelo clasificador utilizando un algoritmo de aprendizaje. En esta etapa el modelo se construye tomando en cuenta la correspondencia de los datos de entrada (en el ejemplo es la edad y el ingreso mensual) con la salida (en el ejemplo es el tipo de auto) que en términos de clasificación corresponde a la clase conocida.

Una vez que se ha construido el modelo clasificador la siguiente etapa es la evaluación, en esta etapa se utiliza el conjunto de datos de evaluación. El objetivo de esta etapa es utilizar el modelo clasificador de la etapa anterior con nuevos datos, en esta etapa el conjunto de datos no contiene la clase, ya que el objetivo es que el modelo asigne esta clase para posteriormente comprar con las clases reales.

Una vez que se generó el modelo clasificador puede ser comparado con otros modelos para saber cuál es la mejor opción según el tipo de problema de clasificación al que nos enfrentamos.

2.2.5. Reducción de dimensionalidad

Existen diferentes técnicas para el pre procesamiento de los datos que se utilizan en el proceso de clasificación ya que los datos que utilizamos pueden contener valores atípicos, valores que faltan, datos registrados incorrectamente, datos duplicados, etc. Es por esta razón que los datos deben ser tratados antes de realizar las etapas de clasificación.

Una de estas técnicas es la reducción de dimensionalidad, para entender que se realiza en este proceso es importante entender que cuando el tamaño de los datos (número de atributos) es grande o aumenta, la dispersión de los datos también aumenta. Esto tiene como consecuencia un proceso de datos difícil debido a la necesidad de aumento de memoria y calculo en los datos.

Para tratar con un tamaño más pequeño de datos y así reducir tiempo y memoria en el procesamiento de datos se realiza una reducción de dimensionalidad. Esto permite obtener un conjunto de datos que pueda ser procesado más rápido y donde se eliminen características irrelevantes (Gorunescu F. , 2011).

2.2.5.1. Singular Value Decomposition

En esta tesis utilizamos el método SVD (*Singual Value Decomposition*) para la reducción de dimensionalidad de las matrices. A continuación, se explica el método SVD de acuerdo a Manning (2009).

Dada una matriz de *M* términos y *N* documentos llamada *C* donde (salvo una rara coincidencia) $M \neq N$ por lo que es muy poco probable que sea simétrica, sea *U* una matriz de $M \times M$ cuyas columnas son vectores propios ortogonales de CC^T (denotando como C^T una matriz transpuesta de *C*) y *V* una matriz $N \times N$ cuyas columnas son vectores propios ortogonales $C^T C$. Tomando en cuenta lo anterior es posible expresar el siguiente teorema tomado de Manning (2009, pág. 408).

Teorema 1 . *Sea r el rango* $M \times N$ *de la matriz término – documento C. Entonces hay una descomposicion del valor singular de C de la siguiente forma:*

$$C = U\sum V^T,$$

Donde:

a) Los valores propios $\lambda 1, \ldots, \lambda r$ de CC^T son los mismos valores propios de $C^T C$;

b) Para $1 \leq i \leq r$, sea $\sigma i = \sqrt{\lambda_i}$, con $\lambda_i \geq \lambda_{i+1}$. Entonces $\sum$ de lamatriz $M\ xN$ es compuesta por $\sum_{ii} = \sigma_i$, para $1 \leq i \leq r$, y cero en caso contrario.

Los valores $\boldsymbol{\sigma_i}$ son los valores singulares de *C*. Entonces ∑ es representada como una matriz de $r \times r$ con los valores singulares en las diagonales y los valores fuera de las diagonales como cero.

La técnica SVD se ha utilizado para obtener una aproximación de bajo rango C_k de *C*. Donde *k* es un entero positivo más pequeño que el rango *r*. Esta aproximación de bajo rango es obtenida derivando $\sum_k$ de ∑ remplazando los valores singulares $r - k$ más pequeños por cero. Entonces C_k es calculado como $C_k = U \sum_k V^T$. Esta aproximación sigue la idea de que el efecto del producto de los valores propios pequeños en la matriz es pequeño.

2.3.Recuperación de información

Una de las principales tareas de la minería de texto es la recuperación de información, esta tarea es asociada principalmente a los documentos en línea y los motores de búsqueda. Un ejemplo típico de recuperación de información es el uso de un motor de búsqueda en donde la entrada son ciertas palabras y la salida son documentos que coinciden con las palabras.

La Figura 2-7 representa el proceso de recuperación de información en donde se obtiene una colección de documentos de acuerdo a ciertas palabras que ayudan a identificar el contenido de los documentos, estos documentos son la respuesta a una consulta. En este ejemplo tenemos como entrada un documento, el objetivo es recuperar otros documentos de una colección que tengan relación con el documento de entrada.

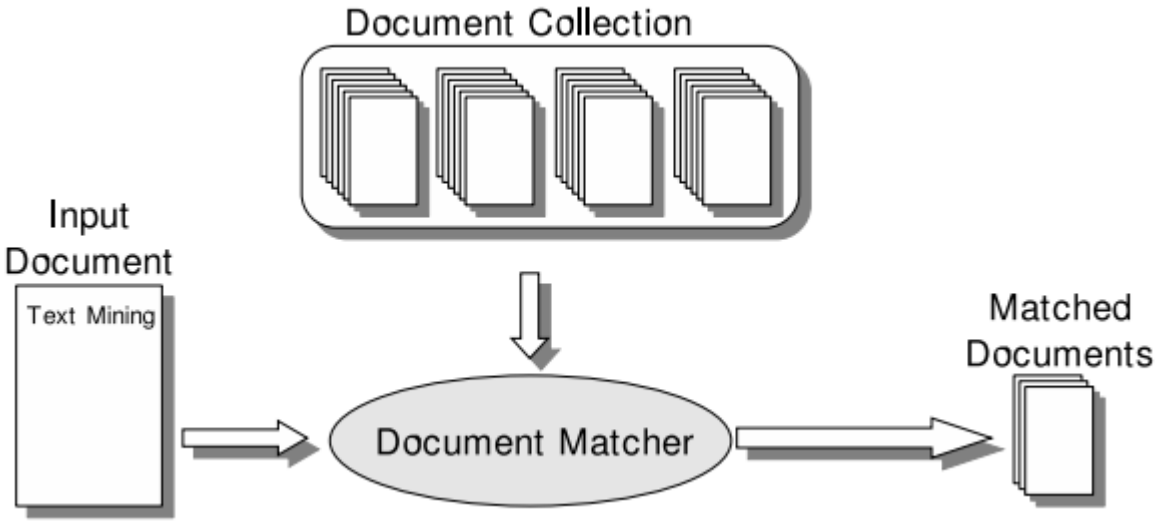


Figura 2-7 Recuperación de información

Tomada de Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F. (2010, pág. 8)

2.3.1. Definición

La recuperación de información es la tarea que trata la búsqueda automática de material relevante (por lo regular son documentos) en grandes colecciones de datos no estructurados (por lo regular es texto) que satisface una necesidad de información (Manning, Raghavan, & Schütze, 2009).

En la Figura 2-8 se presentan las etapas de la recuperación de información. El primer paso es tener una consulta específica, generalmente puede ser una o varias palabras clave. Después se realiza una búsqueda en una colección de documentos y finalmente el sistema regresa documentos que son relevantes a la consulta (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010).

Figura 2-8 Etapas de la recuperación de información

Tomada de Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F. (2010, pág. 85)

2.3.2. Espacio vectorial

Como habíamos mencionado anteriormente, en las tareas de minería de texto se pueden aplicar métodos de minería de datos, la diferencia es que en la minería de textos los datos se encuentran no estructurados, generalmente es texto sin formato por lo que es necesario convertir el texto a un formato estructurado para que pueda ser procesado, en este caso, por los sistemas de minería de texto y específicamente de recuperación de información.

La forma de convertir el texto a un formato estructurado es creando un espacio vectorial (matriz) de palabras o términos de documento. Manning, Raghavan, & Schütze (2009) nos proponen un ejemplo que se muestra en la Figura 2-9. Es la representación de una matriz binaria de terminos de documento de una colección de libros de Shakespeare (colección de documentos), esta matriz se contruye basada en la aparicion de las palabras en un texto. Por ejemplo, en el documento de *Hamlet* se encuentra la palabra *mercy,* por lo que el valor en la matriz es 1. De esta forma se genera un vector para cada término que muestra los documentos en los que aparece o un vector por cada documento que muestra los términos que contiene.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	…
Antony	1	1	0	0	0	1	
Brutus	1	1	0	1	0	0	
Caesar	1	1	0	1	1	1	
Calpurnia	0	1	0	0	0	0	
Cleopatra	1	0	0	0	0	0	
mercy	1	0	1	1	1	1	
worser	1	0	1	1	1	0	
…							

Figura 2-9 Matriz de términos de documento

Tomada de Manning, Raghavan, & Schütze (2009, pág. 4)

2.3.3. Medida TF-IDF

Para construir los vectores de términos de documentos los valores pueden ser establecidos de acuerdo a la frecuencia de aparición de una palabra en un documento, pero también hay que tomar en cuenta que un documento o una parte del documento donde menciona el término de la consulta más a menudo, puede tener más relación con la consulta, por lo tanto, debe tener una puntuación más alta que los demás términos. Con este fin se asigna un peso a cada termino en un documento que depende del número de apariciones en el documento (Manning, Raghavan, & Schütze, 2009).

El método para calcular la frecuencia de aparición de palabras en un documento se llama frecuencia de los términos (*Term frequency*) y asigna el peso igual al número de ocurrencias de los términos *(t)* en un documento *(d)* y se denota como $TF_{t,d}$ con los subíndices (t, d) que indican el término y el documento respectivamente. Si utilizamos esta medida únicamente tendremos información del número de palabras en el documento, por ejemplo, si solo utilizamos la frecuencia de documento, en el documento "María es más rápida que Juan", en cierto modo sería igual al documento "Juan es más rápido que María" ya que ambos documentos contienen la misma frecuencia de palabras, sin embargo, puede ser que ambos documentos no sean relevantes para una misma consulta.

Para Manning, Raghavan, & Schütze (2009) la frecuencia de los términos tiene un problema ya que todos los términos son considerados de igual importancia al evaluar la relevancia de consulta. Por ejemplo, en una colección de documentos sobre la industria automovilística es muy probable que el termino auto esté presente en casi todos los documentos. Con el fin de disminuir el efecto de los términos que se producen con demasiada frecuencia se utiliza otra medida denominada *Inverse document frequency.*

Para obtener la frecuencia inversa de documento se toma en cuenta el número total de documentos de la colección *(N)* y la medida de frecuencia de documentos (*document frequency*) que se denota como df_t y se define como el número de documentos en la colección que contienen un término *(t)*.

La frecuencia inversa de documentos se denota como idf_t con subíndice *t* que significa el término y se obtiene de la siguiente manera:

$$idf_t = \log \frac{N}{df_t}$$

La medida TF-IDF (*Term frequency – Inverse document frequency*) es una combinación de los términos de frecuencia (*Term frequency*) y la frecuencia inversa de documentos (*Inverse document frequency*) y se obtiene de la siguiente manera

$$TF - idf_{t,d} = tf_{t,d} \times idf_t$$

La medida TF-IDF es:

1. Más alta cuando el término *(t)* ocurre muchas veces dentro de un pequeño número de documentos (Dando así la máxima capacidad de discriminación de esos documentos).
2. Baja cuando el termino ocurre (aparece) un menor número de veces en un documento, o aparece en muchos documentos (documentos que ofrecen así una señal de relevancia menos pronunciado)
3. Más baja cuando el término se utilice prácticamente en todos los documentos.

2.3.4. Evaluación

Existen diferentes medidas para evaluar un sistema de recuperación de información, las más frecuentes son *precision*, *recall, accuracy* y *F-score.* Estas medidas de evaluación se explican más a detalle en la sección 3.6.1.

3. Recuperación automática de anécdotas

Como planteamos al inicio de esta tesis, uno de los dos objetivos principales es la detección automática de anécdotas a partir de los comentarios generados en la página de Facebook. Detectar este tipo de narraciones ayudaría a los administradores de *La ciudad de México en el tiempo* a crear un tipo de memoria colectiva. De hecho, como ya se dijo, en diversas ocasiones se les ha pedido ejemplos de recuerdos y anécdotas para ser publicados en libros. Además, tratar de detectar este tipo de narraciones resulta interesante desde la perspectiva del análisis automático de textos en redes sociales.

En los siguientes apartados se describen los pasos realizados en la investigación de un método para detectar anécdotas en comentarios de Facebook. Nos basamos en la metodología del aprendizaje automático, especialmente de la clasificación automática, para llevar a cabo la detección de anécdotas. En general, esta metodología parte de un conjunto de datos ya clasificados que son analizados automáticamente por un algoritmo de aprendizaje para generar un modelo clasificador. Luego, usando datos de evaluación (también ya clasificados) se determina el mejor modelo dadas distintas combinaciones de parámetros y características.

3.1.Recopilación del corpus de comentarios

Para iniciar con la tarea de aprendizaje automático el primer paso es la recolección de los datos para formar un corpus, es decir, una colección de documentos. En este caso buscamos recolectar comentarios[2] de la página de Facebook *La ciudad de México en el tiempo,* para ello nos *centramos* únicamente en los comentarios realizados a las imágenes publicadas en el sitio.

Para formar el corpus, lo primero que se realizó fue conocer la estructura de la página *La ciudad de México en el tiempo*. En ella se encuentran diferentes álbumes de fotografías y cada álbum tiene cierta cantidad de comentarios de diferentes usuarios de *Facebook*. En total se encuentran 29 álbumes y el número de imágenes en cada uno de ellos aumenta cada día. Los nombres de los álbumes de fotografías se presentan enseguida.

- Videos
- Fotos de biografía
- Cargas desde celular

[2] Desde la recuperación de información es posible ver a cada comentario como un documento.

- Catálogo de juguetes Lili Ledy 1979
- Fotos de perfil
- Catálogo de juguetes Lili Ledy 1978
- La ciudad en blanco y negro parte tres
- Desde las alturas- Fotos panorámicas de la Ciudad
- La ciudad a color parte dos
- La ciudad en blanco y negro parte cinco
- El metro
- Pasado y presente
- La Ciudad Universitaria
- Desde las alturas parte dos
- Pulquerías y cantinas
- Escuelas de la ciudad de México
- La ciudad a color
- La ciudad en blanco y negro parte cuatro
- La ciudad en blanco y negro parte dos
- La ciudad en blanco
- 1968
- Cines y teatros
- Transportes parte dos
- Transportes
- Terremoto de 1985
- Litografías, grabados, pinturas y mapas de la ciudad
- Edificios abandonados y más
- Terremoto de 1957
- Fotos de portada

La tarea en esta etapa es obtener los comentarios de cada uno de los álbumes en documentos de texto para su posterior análisis. Para lograrlo se ha utilizado *FQL* y el lenguaje de programación Python. Entre las herramientas para el desarrollo de aplicaciones web y móviles que ofrece *Facebook* se encuentra *FQL (Facebook Query Languaje)* que es un lenguaje que permite la consulta de las tablas (usuarios, post, amigos, imágenes, comentarios, etc.) de la base de datos de *Facebook*, entregando la información en formato Json.

La consulta de estos datos se puede realizar a través de la URL indicando la tabla y los atributos que buscamos, la sintaxis es similar a la del lenguaje SQL. Por ejemplo, en la siguiente consulta buscamos el atributo *objet_id* de la tabla *photo* donde el atributo *aid* sea igual a "187533597935335_43100" que es el id del álbum que queremos consultar.

```
select object_id from photo where aid="187533597935335_43100"
```

Para realizar esta consulta a través de un URL se coloca la instrucción anterior de la siguiente forma:

```
http://graph.facebook.com/fql?q=select object_id from photo
where aid="187533597935335_43100"
```

El resultado de la consulta es en formato Json. Un ejemplo de esta salida se puede ver en la Figura 3-1.

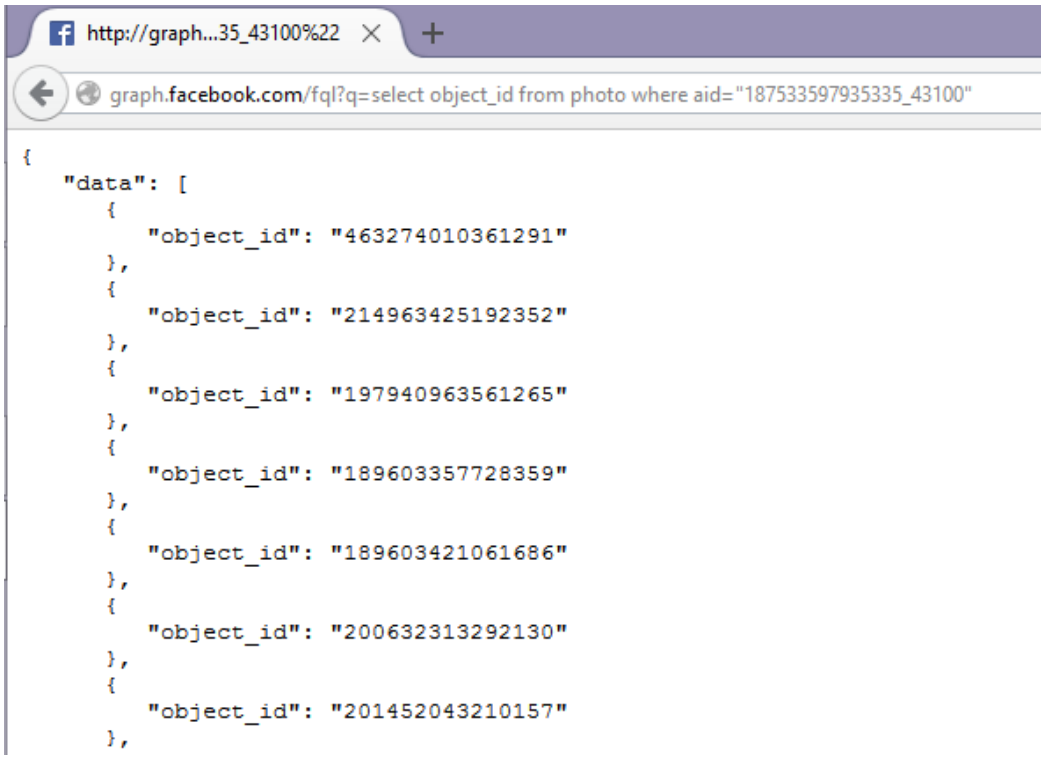


Figura 3-1 Salida de una consulta en formato Json

Para conocer cada una de las tablas y atributos de la base de datos de Facebook se puede consultar la documentación oficial de FQL (https://developers.facebook.com/docs/technical-guides/fql).

Además de poder realizar las consultas a través de la URL en un navegador, también se pueden hacer peticiones en cualquier lenguaje de programación que tenga una librería URL. En

este trabajo de investigación se ha utilizado la librería urllib2 de Python para obtener los comentarios de cada fotografía, los cuales fueron almacenados en un archivo de texto.

Para identificar la imagen a la que se refiere el comentario, se nombró a cada archivo con el *Facebook ID* que es un identificador único que asigna Facebook a cada objeto, en este caso las fotografías.

3.2.Etiquetado lingüístico del corpus

Considerando que uno de los objetivos de esta tesis es la detección de anécdotas y que estas se pueden definir como un relato breve de un suceso ocurrido, es lógico pensar que en una anécdota encontraremos verbos en tiempo pasado. Esto nos llevó a distintas intuiciones para la detección de anécdotas, las cuales requerirían de marcar los verbos dentro de los comentarios. Por lo tanto, una de las características que se tomarán en cuenta para generar modelos de clasificación es la categoría gramatical (parte de la oración) a la que pertenece cada palabra. Es decir, será necesario determinar si cada palabra es un:

- Sustantivo
- Adjetivo
- Articulo
- Pronombre
- Verbo
- Adverbio
- Conjunción o
- Preposición

Esta tarea de asignar a cada palabra de un corpus su categoría gramatical se conoce como etiquetado de partes de la oración (*Part-Of-Speech Tagging, POST)*. El POST es una tarea común en la minería de textos (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010, págs. 37-39). Para obtener las categorías gramaticales, el corpus fue procesado con la herramienta FreeLing[3] que es una herramienta que se encarga de asignar etiquetas a cada palabra según la categoría a

[3] http://nlp.lsi.upc.edu/freeling/

la que pertenece. Las etiquetas asignadas por esta herramienta siguen la convención del *Expert Advisory Group on Language Engineering Standards*[4] (EAGLES).

Esta convención ha propuesto un conjunto de etiquetas para representar la categoría gramatical de cada palabra. Para formar estas etiquetas se toman en cuenta diferentes atributos lingüísticos como el género, persona, número, etc. Cada uno de estos atributos tiene una posición, por ejemplo, en la posición número 1 siempre se encuentra el atributo *categoría*, el valor de este atributo se establece dependiendo de la palabra que se analice, por ejemplo, un verbo es representado con el código V, adjetivo con el código A, artículo con el código T, etc.

Una vez que se ha establecido la categoría, en las siguientes posiciones se encuentran atributos de esa categoría, por ejemplo, en verbo podemos encontrar tipo, tiempo, persona, etc. y cada uno de estos valores tiene un código. Tomando en cuenta los códigos de estos atributos se forman las etiquetas que representan la categoría gramatical de la palabra. En la Tabla 3-1 se muestran las posiciones, atributos, valores y códigos que puede tomar un verbo[5].

VERBOS			
Posición	**Atributo**	**Valor**	**Código**
1	Categoría	Verbo	V
2	Tipo	Principal	M
		Auxiliar	A
3	Modo	Indicativo	I
		Subjuntivo	S
		Imperativo	M
		Condicional	C
		Infinitivo	N
		Gerundio	G
		Participio	P
4	Tiempo	Presente	P
		Imperfecto	I
		Futuro	F
		Pasado	S
5	Persona	Primera	1
		Segunda	2

[4] http://www.ilc.cnr.it/EAGLES/home.html

[5] http://www.cs.upc.edu/~nlp/tools/parole-sp.html

		Tercera	3
6	Número	Singular	S
		Plural	P
7	Género	Masculino	M
		Femenino	F

Tabla 3-1 Etiquetas para verbos

Considerando la Tabla 3-1 podemos definir las etiquetas para distintos verbos, por ejemplo, para el verbo **canté** se toma en cuenta el código de cada posición formando la etiqueta VMIS1S0, que significa: Verbo principal en modo indicativo, tiempo pasado de la primera persona de singular. La Tabla 3-2 muestra esta información con sus etiquetas respectivas.

Posición	Atributo	Valor	Código
1	Categoría	Verbo	V
2	Tipo	Principal	M
3	Modo	Indicativo	I
4	Tiempo	Pasado	S
5	Persona	Primera	1
6	Número	Singular	S
7	Género		0

Tabla 3-2 Etiqueta para verbo en pasado canté

Para el corpus con etiquetado lingüístico (POST), procesamos cada uno de los comentarios con la herramienta FreeLing la cual asigno etiquetas para cada una de las palabras. Por ejemplo, en el comentario:

"yo fui todavia en los ochentas"

Las etiquetas para cada una de las palabras son:

Yo/PP1CSN00 fui/VMIS1S0 todavia/NCFS000 en/SPS00 los/DA0MP0 ochentas/NCFP000

Para conocer la categoría gramatical a la que pertenece cada etiqueta se pude consultar la documentación de FreeLing[6]. Nótese como el adverbio *todavía* fue etiquetado erróneamente como sustantivo común masculino singular (NCFS000) por la falta de acento. Este es uno de los retos de esta tesis, ya que, por ser comentarios de una red social, no se siguen convenciones ortográficas.

El resultado del etiquetado lingüístico fue el mismo corpus, pero representado con etiquetas que indican las partes de la oración. Por lo tanto, tenemos dos versiones del corpus para realizar experimentos y evaluación.

3.3.Clasificación manual

Para la tarea de aprendizaje automático supervisado, es necesario establecer previamente un conjunto de datos de entrenamiento, es decir, la máquina necesita tener datos para aprender y posteriormente realizar clasificación automática en nuevos datos. Lo anterior conlleva obtener un conjunto de características de cada una de las clases y generar un modelo de clasificación que sea capaz de predecir estas clases.

Teniendo en cuenta lo anterior, la etapa de clasificación manual consiste en asignar manualmente una etiqueta a cada documento dependiendo si es anécdota o no es anécdota (clasificación binaria). Las clases que se han establecido son *Anécdota* y *No anécdota*, que se etiquetaron con las etiquetas SI y NO, respectivamente. Para la clase Anécdota, consideramos los comentarios que son anécdotas o recuerdos tomando en cuenta las siguientes definiciones obtenidas del diccionario de la Real Academia Española:

- Anécdota: Relato breve de un hecho curioso que se hace como ilustración, ejemplo o entretenimiento.
- Recuerdo: Memoria que se hace o aviso que se da de algo pasado o de que ya se habló.

Para realizar esta tarea de clasificación manual se recibió apoyo de parte de alumnos del servicio social del Grupo de Ingeniería Lingüística que estudian las carreras de Informática y Lengua y Literaturas Hispánicas. Se dividió el corpus en 10 partes y a cada alumno se le asignó una parte con el fin de que asignara una etiqueta a cada comentario.

[6] http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html

En el corpus se encuentran diferentes comentarios que cumplen con las características para ser consideradas anécdotas o recuerdos; a estos se les asignó la etiqueta SI. En este punto es importante mencionar que para asignar la etiqueta se les pidió que modificaran el nombre del archivo colocando la etiqueta de clase al inicio de este (SI o NO). Por lo tanto, el nombre de los comentarios quedo de la siguiente forma SIc_636213073067383_2250300.txt[7]. Es importante mencionarlo, ya que el nombre del archivo se tomará en cuenta en las siguientes etapas para determinar la clase a la que pertenece.

A continuación, se muestran comentarios como ejemplos de anécdotas o recuerdos:

> "Cuando era niño fui testigo que cuando este canal tenía el agua limpia varios niños de la primaria Gabino barreda que está en la misma calle se metían a nadar ¡un pequeño balneario en pleno centro histórico! bonitos recuerdos de mi niñez hoy a mis 40 años".

> "Fui parte de la historia de ese momento, mas no fui parte del movimiento estudiantil. Fui músico rocanrolero, tocaba el órgano con un grupo de rock llamados los shippy's en 1968 y tocábamos en un café cantante de la zona rosa llamado 2+2 que se encontraba en la esquina de las calles de Londres y Amberes. ¿Por qué estábamos ahí? por puro accidente. Fuimos a recoger a dos chicas al edificio chihuahua para ir al cine y cenar. Lo que vimos desde el cuarto piso de ese edificio no lo olvidaré nunca, fue horriblemente increíble y triste, me sentí incrédulo de ver entrar a los soldados por el lado de reforma disparando y viendo a la gente caer y después por el lado del cine Tlatelolco entraron más soldados haciendo lo mismo, disparando. Luego aparecen las tanquetas detrás de la vieja iglesia y disparan al edificio donde estábamos".

> "yo naci julio 27 de 1957 en medio de un temblor y el angel se desplomo. en 1970 la oficina de mi padre ocupo el piso 13 de reforma 364 mejor conocido como el edificio bch actualmente univicion mexico ocupa el espacio de la oficina como estudio. cada vez que veo esto por televicion me trae un sin numero de memorias de mi magnifico padre emilio r ebergenyi belgodere importantisima persona en el desarrollo industrial del pais."

Por otra parte, en el corpus se encuentran comentarios que no cumplen con las características para ser considerados anécdotas o recuerdos. Estos comentarios pertenecen a la clase de No anécdotas y se les asigna la etiqueta NO de la misma forma que ya se ha descrito anteriormente. A continuación, se ejemplifican algunos comentarios de la clase No anécdota:

> "muchas, muchas gracias por estas imágenes que nos regalas y que nos hacen apreciar más nuestra increíble y maravillosa ciudad :)".

[7] En el ejemplo, el primer conjunto de dígitos (636213073067383) se refieren al ID de la fotografía asignado por Facebook y el segundo (2250300) es el ID del comentario.

> “ay q nostalgia triste de lo q hemos hecho a nuestra tierra. Adiós canal de la viga, adiós rio Mixcoac, adiós rio becerra. Hola cemento, concreto y caos”.

> “en esa foto se ve terrorifico el museo de cera qon esta entrada nada que ver qon aorita jajajjaja parece qomo la peli de terror de la qasa de cera jajajaja visitenlo vale la pena ir”

Como puede observarse, este tipo de textos incluye diversos retos para su procesamiento automático que provienen del hecho de ser extraídos de redes sociales. Entre ellos se pueden observar errores ortográficos (nací > “naci”, México > “mexico”, televisión > “televicion”), acortamiento de palabras (que > “q”), uso de emoticones (“:)”), uso de prácticas de escritura propia de las redes sociales (con, como, casa > “qon, qomo, qasa”) y un tipo de narración cercana a la oral.

Una vez finalizada la clasificación manual asignando las etiquetas SI y No respectivamente, el número de comentarios del corpus se redujo ya que del total de comentarios recolectados en un inicio únicamente se encontraban 1,405 comentarios de la clase Anécdota. Es por esto que se eligieron aleatoriamente 1,405 comentarios pertenecientes a la clase No anécdotas con el fin de equilibrar el corpus y obtener la misma cantidad de comentarios de cada clase obteniendo un total de 2,810 comentarios[8].

Es importante recordar que en la tarea de clasificación automática son necesarias, entre otras, las etapas de entrenamiento y evaluación. Teniendo en cuenta estas dos etapas y una vez obtenido el corpus clasificado manualmente, el siguiente paso fue dividir el corpus para las etapas de entrenamiento y evaluación. El corpus quedo dividido de la siguiente manera.

De los 2,810 comentarios se dividió en porcentajes de 70% para la etapa de entrenamiento y 30% para la etapa de evaluación. Por lo tanto, el corpus de entrenamiento tiene un total de 1,966 comentarios y el corpus de evaluación tiene un total de 844 comentarios. Es importante tomar en cuenta que la selección de comentarios para cada corpus se realizó de forma aleatoria, pero respetando la misma cantidad de comentarios de cada clase (Anécdota/No anécdota) en cada corpus. A continuación se presenta una tabla (Tabla 3-3) de la distribución de los comentarios en cada corpus.

[8] Aunque decidimos utilizar un corpus equilibrado, estamos conscientes de que también podíamos haber utilizado un corpus que reflejara la distribución desequilibrada de las clases. Dado que esta tesis es un primer acercamiento a la detección de anécdotas, preferimos usar un corpus equilibrado para evitar que asignando la clase más frecuente se obtuvieran mejores resultados.

Total de comentarios			
2,810 (100%)			
Entrenamiento		Evaluación	
1,966 (70%)		844(30%)	
Anécdota	No anécdota	Anécdota	No anécdota
983	983	422	422

Tabla 3-3 Distribución de comentarios en corpus de entrenamiento y evaluación

3.4.Entrenamiento

En las etapas anteriores hemos recopilado, realizado el etiquetado manual y dividido el corpus para utilizarlo en las etapas de entrenamiento y evaluación, además, también realizamos el etiquetado lingüístico al corpus con el fin de realizar experimentos tomando en cuenta diferentes categorías gramaticales.

Las siguientes etapas son de entrenamiento y posteriormente la evaluación. La tarea principal de la etapa de entrenamiento en este caso es proporcionar a la computadora las características que debe poseer un comentario para formar parte de una de las dos clases que hemos establecido (Anécdota y No anécdota). Para esta tarea se toma en cuenta únicamente el corpus de entrenamiento.

Para esta tarea hemos utilizado la librería *Scikit-learn*[9] en el lenguaje de programación Python. Dicha librería provee diferentes herramientas para las tareas del procesamiento del lenguaje natural, aprendizaje automático, minería de datos, análisis de datos, etc. Entre estas tareas se encuentra el aprendizaje automático supervisado y en específico provee algoritmos para realizar clasificación automática.

Para llevar a cabo las siguientes etapas realizamos un programa que hemos nombrado entrenarEvaluar.py. Dicho programa tiene como objetivo generar un vectorizador y un modelo clasificador de comentarios donde las clases a identificar son Anécdota y No anécdota, además, el programa permite realizar experimentos de clasificación con diferentes características de los textos.

Con programa entrenarEvaluar.py es posible realizar entrenamiento y posteriormente la evaluación, para realizar esto se toma en cuenta los corpus de entrenamiento y evaluación. La

[9] http://scikit-learn.org/

salida de este programa son diferentes archivos: un vectorizador, un modelo clasificador y un reporte con los resultados de la ejecución del programa. Este reporte incluye los parámetros de ejecución, los mejores parámetros obtenidos por el grid_search para el algoritmo utilizado, el resultado de las medidas de *accuracy* para la etapa de entrenamiento y para la etapa de evaluación, también las de *precision*, *recall* y *F-score* (solo para evaluación). Adicionalmente se muestra la matriz de confusión y un resumen de clasificación[10].

3.5.Preprocesamiento

Utilizando las herramientas mencionadas anteriormente, obtuvimos archivos en formato Json. En estos archivos existe contiene información que no es necesaria para la tarea de clasificación automática, por ejemplo, id de la imagen, numero de *likes,* fecha de publicación, id del usuario que realizo el comentario, etcétera[11].

Como nuestro objetivo es la recopilación del corpus de comentarios, los archivos fueron procesados con el lenguaje de programación Python para obtener únicamente los textos de los comentarios de cada imagen (archivos de texto plano). Cada comentario fue almacenado por separado en un archivo diferente.

3.5.1. Experimentos

Con el fin de obtener un modelo clasificador de Anécdotas y No Anécdotas se realizaron diferentes experimentos en la etapa de entrenamiento en donde se tomaron en cuenta diferentes características de los comentarios. Dichos experimentos se realizaron tomando en cuenta el corpus de entrenamiento que contiene únicamente el texto y el corpus que contiene etiquetas de acuerdo a la categoría gramatical. Esto se realiza con el fin de generar diferentes modelos clasificadores y elegir el mejor de ellos. A continuación se presentan las características a tomar en cuenta en cada corpus. Todos los textos son regularizados a letras minúsculas.

Características para el corpus de entrenamiento que contiene únicamente texto:

- Uso de *stop words*: Se refiere a las palabras que se encuentran con mucha frecuencia en el español, por ejemplo preposiciones, pronombres, artículos,

[10] Los detalles de este reporte, el cálculo de las medidas y la descripción de la matriz se explicarán en la sección 3.6.1.

[11] Esta información podría ser útil en otro tipo de tarea de procesamiento de lenguaje natural, pero en nuestro caso consideramos que no sería útil.

adverbios, etc., que no aportan un significado al texto, se realizaron experimentos usando estas palabras y eliminando estas palabras de los textos.

- Uso de acentos: Se realizaron experimentos con acentos y eliminando los acentos
- Truncar palabras: Se refiere a que cada palabra del texto se trunca a 6 letras, se realizaron experimentos con palabras completas y truncando palabras. El truncamiento de palabras permite asociar diversas variantes ortográficas a un solo representante truncado (proceso de normalización de palabras). Por ejemplo, las palabras "historia, historias, historiador" se representan por la palabra truncada "histor".
- Número de N-grama: Se refiere a *n* número de elementos en este caso de palabras. Por ejemplo de la frase "Fui parte de la historia de ese momento" los bigramas (*n* = 2) obtenidos serían: "Fui parte", "parte de", "de la", "la historia", "historia de", "de ese", "ese momento".

Características para el corpus de entrenamiento que contiene etiquetas:

- Categoría gramatical: Se realizaron experimentos tomando en cuenta todas las categorías gramaticales que se encontraron en los textos.
- Verbos: Del texto con etiquetas se tomaron en cuenta únicamente las etiquetas que comienzan con la letra V que se refiere a todas las palabras que se encuentran dentro de la categoría Verbo.
- Verbos en pasado: Se tomaron en cuenta las etiquetas que comienzan con VMIS que pertenecen a la categoría pretérito perfecto simple (pasado).
- Verbos en copretérito: Se tomaron en cuenta las etiquetas que comienzan con VMII que pertenecen a la categoría pretérito imperfecto (copretérito).

En general, para los dos corpus se tomó en cuenta:

- Algoritmo: Se utilizó el algoritmo SVM en todos los experimentos y se seleccionaron algunos para el algoritmo Naïve Bayes con el objeto de comparar resultados. Para el algoritmo SVM se variaron tres parámetros: el tipo de kernel (rbf, polynomial y linear), el valor C (1, 10, 100 y 1000) y el valor gamma (solo se probaron 0.001 y 0.0001). Se probaron tres tipos de algoritmos Naïve Bayes: Gaussian, Multinomial y Bernoulli; solo para los dos últimos se variaron los valores alpha (0, 0.01 y 1).

- Tipo de vectorizador (valores de las características): Para realizar la clasificación automática los comentarios se almacenan en forma de un matriz (conjunto de vectores) en donde cada una de las columnas es un término (palabra o n-grama) con las características que mencionamos anteriormente como acentos, *stop words* y palabras truncadas. En el caso del corpus con etiquetado lingüístico, cada columna es una etiqueta de la categoría gramatical. Las filas de la matriz (vector) corresponden a cada uno de los comentarios del corpus (documento). A continuación se muestra una representación de este tipo de matrices (Tabla 3-4)

	$término_1$	$término_2$	$término_3$	$término_4$	$término_5$	…	$término_m$
doc_1	$valor_{1,1}$	…	…	…	…	…	$valor_{1,m}$
doc_2	…	$valor_{2,2}$	…	…	…	…	…
doc_3	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…
doc_n	$valor_{n,1}$	…	…	…	…	…	$valor_{n,m}$

Tabla 3-4 Representación de matriz para clasificación automática

Es importante recordar que en esta tarea cada comentario puede ser llamado también documento. A continuación se explican cada uno de los vectores que se tomaron en cuenta en los experimentos y que proporcionaron cada uno de los valores ($valor_{n,m}$) de la matriz.

- Binario (BINAR)

  El vector binario tiene los valores de 0 o 1, si el término de la columna aparece en el comentario el valor que toma es 1 de lo contrario el valor es 0.

- Conteo (COUNT)

  En el vector de conteo, cada elemento de la matriz tiene un valor dependiendo el número de veces que aparece el término de la columna en un comentario.

- TF-IDF (TFIDF)

  En el vector TF-IDF cada elemento tiene el valor tf-idf del término. TF-IDF es una medida que representa la relevancia de un término en un documento (en este caso en un comentario) que se encuentra una colección de documentos (en todo el corpus).

- Reducción de dimensionalidad

  Una opción que tenemos al momento de crear una matriz para clasificación automática es la de reducir la dimensionalidad, es decir, reducir el número de características (columnas). En esta tesis solo se utilizó el método de *Singular Value Decomposition* (SVD) para reducir la dimensionalidad de las matrices de comentarios.

Considerando estas características hemos realizado diferentes experimentos utilizando el programa entrenarEvaluar.py. La forma de ejecutar el programa es mediante una línea de comandos, en esta tesis realizamos experimentos en los sistemas operativos Windows y Linux. Las herramientas que utilizamos son Python 3.4 y las bibliotecas *Scikit-learn, NumPy* y *SciPy* del mismo Python.

Parámetro	**Descripción**
-d	Indica la ruta del directorio donde se encuentra el corpus
-p	Utilizar etiquetas POS (corpus con etiquetado lingüístico)
-e	Expresión regular para indicar que etiquetas POS serán utilizadas
-i	n-grama inicial, se indica con número entero
-f	n-grama final, se indica con número entero
-v	Indica el tipo de vectorizador. Las opciones son TFIDF,COUNT o BINAR
-s	Indica si se filtran *stop words* en el texto
-a	Eliminación de acentos. Las opciones son unicode, ascii o none (no eliminar)
-r	Truncar palabras
-m	Número de componentes (columnas) resultado de la reducción de dimensionalidad
-t	Transformación para reducción de dimensionalidad. Opciones: SVD, PCA-gauss, PCA-poly2, PCA-poly3 (solo se utilizó SVD)
-l	Algoritmo de clasificación, Las opciones son SVM, MultinimialNB y otros
--normalizacion	Indica el tipo de normalización de los valores de la matriz. Las opciones son log2 y None
-u	Tipo de kernel, únicamente para el algoritmo SVM. Las opciones son linear y rbf
--cvalues	Opciones para el valor C, únicamente para el algoritmo SVM. Ejemplo: '1,10,100,1000'

Tabla 3-5 Parámetros de entrada del programa entrenarEvaluar.py

Tomando en cuenta los parámetros de entrada al programa, que se pueden ver en la Tabla 3-5, un ejemplo de ejecución es:

```
python entrenarEvaluar.py -d "G:\comentarios\clasificacion\"
-v TFIDF -l SVM -a unicode -r -m 300 -t SVD -u rbf --cvalues
'1,10,100,1000' --normalizacion None
```

En el ejemplo anterior el corpus se encuentra en la ruta "G:\comentarios\clasificacion\". Además, el programa creará una matriz con valores TF-IDF eliminando los acentos con el parámetro Unicode en la opción –a y truncando las palabras (-r). Esta llamada al programa utilizará el algoritmo SVM (-l) con un kernel rbf (-u) y los valores 1,10,100,1000 para C. Finalmente, con los parámetros –t y –m se indica que se realice una reducción de dimensionalidad a 300 columnas.

En el siguiente ejemplo se muestra la forma de ejecutar el programa para utilizar el corpus con etiquetado lingüístico.

```
python entrenarEvaluar.py -d "G:\comentarios\clasificacion\"
-p -e '.+' -v BINAR -a SVM -u rbf --cvalues '1,10,100,1000'
--normalizacion None
```

En este ejemplo se creará una matriz con valores binarios (-v BINAR) utilizando un algoritmo SVM. La opción –p indica que se utilizará el corpus con etiquetas lingüísticas. En este punto es importante mencionar que para realizar los experimentos con el corpus de etiquetas hemos utilizado expresiones regulares para indicar que categoría gramatical usar. El parámetro –e con el valor '.+' es una expresión regular que indica utilizar todas las etiquetas.

En esta tesis realizamos experimentos utilizando verbos para lo cual utilizamos el corpus con etiquetado lingüístico y, como mencionamos en la sección 3.2, cada categoría gramatical tiene una etiqueta y cada documento del corpus está conformado por diferentes etiquetas. Entonces, para indicarle al programa que utilizaremos ciertas etiquetas hemos utilizado diferentes expresiones regulares en Python como se muestra en la Tabla 3-6.

Expresión regular	**Descripción**
'.+'	Todas las etiquetas
'^V'	Todos los verbos
'^VMII'	Solo verbos en pretérito imperfecto (copretérito)
'^VMIS'	Solo verbos en pretérito perfecto simple (pasado)

Tabla 3-6 Expresiones regulares para identificar verbos con etiquetas EAGLES

3.5.1.1. Baseline

Cuando realizamos los experimentos obtenemos modelos de clasificación que podemos evaluar con diferentes medidas (en este caso evaluamos principalmente con la medida *Accuracy*, que se explicará más adelante); sin embargo, necesitamos una medida base (*baseline*) de la cual partir para saber si el modelo clasificador trabaja mejor que métodos más tradicionales, simples e intuitivos. Entonces el objetivo de realizar diferentes *baseline* es tener una medida que será tomada como punto de referencia para comparar los resultados de los experimentos realizados.

Para obtener estas medidas de *baseline* hemos establecido diferentes métodos para asignar las clases a los comentarios del corpus de evaluación. Una vez establecidas las clases, se han evaluado con la medida *Accuracy* al igual que los experimentos realizados. Los métodos para obtener los *baseline* son:

- Aleatorio: Asigna una clase a cada comentario de forma aleatoria
- Distribución de clases: Asigna una clase a cada comentario de forma aleatoria pero respetando la distribución de clases, por ejemplo, si en nuestro corpus tiene en total 844 comentarios de los cuales 422 (50%) pertenecen a la clase Anécdotas y 422 (50%) a la clase No Anécdotas, este método asignara aleatoriamente 422 clases Anécdotas y 422 clases No Anécdotas.

Además, incluimos algunos métodos *baseline* basado en la intuición de que las anécdotas podrían caracterizarse por el uso de verbos en pasado. Estos métodos son:

- Número de verbos en pasado respecto al número total de verbos: Para este método tomamos en cuenta los corpus de entrenamiento y evaluación con etiquetas. Primero se realiza el conteo de los verbos en pasado (VMIS) y el conteo de todos los verbos (V) de cada comentario del corpus de entrenamiento para obtener la relación de verbos en pasado entre el total de verbos, por lo que se realiza la operación:

  Tot V Pas Simple y Copretérito / Tot V

  Cuando se tiene el resultado de esta operación por cada uno de los comentarios, se obtiene el promedio de estas cantidades al que llamaremos *Media.* Entonces, para asignar clases a los comentarios del corpus de entrenamiento utilizamos las siguientes reglas:

Tot V Pas Simple y Copretérito / Tot V > Media = Anécdota

Tot V Pas Simple y Copretérito / Tot V < Media = No Anécdota

- Número de verbos en pasado respecto al número total de palabras: Al igual que el método anterior se utiliza primero el corpus de entrenamiento para obtener un promedio, pero esta vez obtenemos la relación de verbos en pasado sobre todas las palabras que se encuentran en un comentario utilizando la fórmula:

  Tot V Pas Simple y Copretérito / Tot Palabras

  De la misma forma obtenemos el promedio de estas cantidades y utilizamos las siguientes reglas para asignar clases al corpus de evaluación:

  Tot V Pas Simple y Copretérito / Tot Palabras > Media = Anécdota

  Tot V Pas Simple y Copretérito / Tot Palabras < Media = No Anécdota

Realizamos diferentes experimentos tomando en cuenta estos métodos para obtener medidas *baseline.* Una vez asignadas las clases se obtuvo la medida *accuracy* como se explica en el apartado 3.6.1 Evaluación.

3.6. Evaluación y resultados

En esta sección conoceremos la forma de evaluar los modelos clasificadores que resultaron de los experimentos. Para ello debemos recordar que en la fase de enteramiento se generaron diferentes modelos clasificadores con el corpus de entrenamiento, la evaluación de estos modelos se realizó utilizando el corpus de evaluación.

Como recordaremos, el corpus de evaluación contiene los comentarios con la etiqueta de la clase a la que pertenecen, de esta forma podemos comparar las clases reales con las clases asignadas por el modelo y realizar una evaluación de los diferentes modelos. Además, se debe recordar que estos comentarios son distintos a los utilizados en el corpus de entrenamiento. En los siguientes apartados se explica la forma de evaluación y una discusión de los mejores resultados de clasificación.

3.6.1. Evaluación

Para realizar la evaluación de cada experimento se ha tomado como base la medida de evaluación *accuracy* obtenida de la clasificación sobre el corpus de evaluación. Además de esta medida también hemos tomado en cuenta las medidas *precisión, recall y F-score* que son las

principales medidas de evaluación en la tarea de recuperación de información (Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F., 2010).

Para entender cómo se obtienen estas medidas es importante tomar en cuenta que en esta tarea de clasificación utilizamos un corpus en donde las clases fueron asignadas manualmente, es decir, conocemos las clases de cada comentario y de esta forma podemos realizar una evaluación del modelo clasificador comparando las clases que asignamos manualmente con las clases que asignó el modelo.

Tomando en cuenta lo anterior, se crea una matriz de confusión para representar la comparación entre las clases reales (asignadas manualmente) y las clases predichas (asignadas por el modelo clasificador). De esta forma podemos observar qué tanto se confunde el modelo al asignar las clases. En la Tabla 3-7 se muestra un ejemplo de una matriz de confusión en donde las columnas corresponden a las clases predichas y las filas corresponden a las clases reales, como recordaremos, estamos utilizando dos clases que son Anécdota (clase positiva) y No anécdota (clase negativa). Los valores que se encuentran en la matriz se establecen de acuerdo a la relación entre los dos conjuntos de clases (reales y predichas).

		Predichas	
		Anécdota	No anécdota
Reales	Anécdota	TP	FN
	No anécdota	FP	TN

Tabla 3-7 Estructura de una matriz de confusión

El valor TP corresponde al número de clases *True positive* es decir el número de ejemplos que el modelo asigno correctamente como anécdota, FN corresponde a *False negative,* que es el número de ejemplos que eran anécdotas y el modelo clasificador le asigno incorrectamente la clase No anécdota, FP es el valor *False positive,* es decir, el número de ejemplos que corresponden a la clase No anécdota y el modelo le asigno la clase Anécdota, finalmente se encuentra el valor TN que corresponde al número de clases *True negative,* donde la clase real es No anécdota y el modelo asigno correctamente la clase No anécdota.

Tomando en cuenta los valores de la matriz de confusión se pueden obtener las medidas que mencionamos anteriormente que son *precisión, recall, F-score y accuracy* (Manning, Raghavan, & Schütze, 2009, págs. 154-156), las cuales tomamos en cuenta para comparar los modelos clasificadores. Utilizamos *accuracy* como medida principal para elegir un mejor modelo

clasificador y como veremos a continuación cada una de estas medidas tiene un objetivo diferente por lo que también fueron tomadas en cuenta.

La medida *accuracy* (exactitud) se refiere a qué tan cerca está el modelo de asignar valores verdaderos (TP y TN), es decir, que las clases que asigne a cada comentario sean las correctas, tomando en cuenta la Tabla 3-7, podemos obtener el valor de *accuracy* con la fórmula:

$$Accuracy = \frac{TP + TN}{(TP + FN + FN + TN)}$$

Como ejemplo podemos tomar los valores de la Tabla 3-8 y utilizando la formula, la medida *accuracy* quedaría de la siguiente forma:

Accuracy = 286 + 321 / (286 + 101 + 136 + 321) = 0.71919431

		Predichas	
		Anécdota	No anécdota
Reales	Anécdota	TP :286	FN: 136
	No anécdota	FP: 101	TN: 321

Tabla 3-8 Ejemplo de una matriz de confusión

La siguiente medida que utilizamos en la medida de *precisión*. Esta se encarga de medir la proporción de los documentos recuperados que son relevantes sobre todos los documentos recuperados, en este caso consideramos como documentos relevantes aquellos que en la clase real son Anécdotas y el modelo los clasifico correctamente, es decir el valor TP. La medida *precision* se obtiene con la siguiente formula:

$$Precision = \frac{TP}{TP + FP}$$

Tomando como ejemplo la matriz representada en la Tabla 3-8 podemos obtener la medida de *precision* de la siguiente forma:

Precision = 286 / 286 + 101 = 0.74

Otra medida que tomamos en cuenta es *recall* (exhaustividad) se encarga de medir la proporción de los documentos relevantes recuperados (TP) sobre el total de documentos que

son relevantes en toda las colección de documentos, es decir, el total de documento que en las clases reales pertenecen a la clase Anécdota. La fórmula para obtener esta medida es:

$$Recall = \frac{TP}{TP + FN}$$

Y tomando en cuenta el ejemplo de la Tabla 3-8 la medida *recall* se obtiene de la siguiente forma:

Recall = 286 / 286 + 136 = 0.68

La última medida que tomamos en cuenta es *F-score* que se obtiene calculando la media ponderada de las medidas *presicion* y *recall* para tener un equilibrio entre estas dos medidas, con la siguiente formula:

$$F - score = 2(\frac{Precision \times Recall}{Precision + Recall})$$

Y como ejemplo tomamos los valores de la Tabla 3-8, en donde la medida *F-score* queda de la siguiente manera:

F- score = 2(0.74 * 0.68 /0.74 + 0.68) = 0.71

Por otra parte, los reportes de evaluación que genera el programa fueron analizados para interpretar los resultados de los experimentos con mejor medida de *accuracy*. Como ya se mencionó, estos reportes contienen la matriz de confusión y un resumen de la clasificación. Un ejemplo de un reporte es el siguiente:

```
********** ENTRENAMIENTO **********
Algoritmo: SVM con Kernel: rbf y valores C: 1,10,100,1000
Transformación SVD y componentes: 300
Vectorizador: Vectorizador TFIDF, stop words False, acentos unicode, norma None, ngramas desde 1 hasta 1 truncar<class 'definirVectorizador.truncar'>
Normalización: None
Mejor resultado Accuracy: 0.771617497456765
Mejores parámetros:
     C: 10
     cache_size: 200
     class_weight: None
     coef0: 0.0
     decision_function_shape: None
     degree: 3
     gamma: 0.0001
     kernel: 'rbf'
```

```
     max_iter: -1
     probability: False
     random_state: None
     shrinking: True
     tol: 0.001
     verbose: False
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.8021327014218009
Accuracy: 0.8021327014218009
Precision: 0.8277634961439588
Recall: 0.7630331753554502
F-score: 0.7940813810110975
Matriz de confusión:
[[322 100]
 [ 67 355]]
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.83      0.76      0.79       422
         NO       0.78      0.84      0.81       422

avg / total       0.80      0.80      0.80       844
```

El reporte anterior permite determinar que se obtuvo una medida de *accuracy* de 0.7716 en el entrenamiento, la cual subió para el conjunto de datos de evaluación a 0.8021. Sobre el contenido de los reportes, primero se muestran las opciones de ejecución para el algoritmo: transformación, vectorizador, uso de stop words, uso de acentos, norma, número de n gramas, truncamiento de palabras, normalización. Además, se muestra el mejor resultado *Accuracy* para la fase de entrenamiento.

Después, el reporte muestra los mejores parámetros encontrados por la grid_search para el algoritmo. Se debe recordar que solo se variaron los parámetros del valor C, el kernel y el valor gamma (solo 0.001 y 0.0001). La parte final del reporte presenta los valores obtenidos para el resto de medidas de evaluación y la matriz de confusión.

3.6.2. Resultados

A continuación, se presentan los resultados de los experimentos realizados. Se presentan en forma de tablas en donde se comparan distintos parámetros y algoritmos. Estas tablas solo incluyen los diez resultados más altos. Se brindan además algunas observaciones sobre los mismos. Los resultados de todos los experimentos que se realizaron se muestran en el Anexo 1.

La Tabla 3-9 muestra los mejores resultados obtenidos usando categorías gramaticales para caracterizar los comentarios. Se observa claramente que usar todas las categorías en lugar de solo verbos ofrece mejores resultados, tanto para el algoritmo SVM como para Naïve Bayes (Multinomial). También se observa que los resultados más altos se obtienen con vectores binarios usando solo unigramas de categorías.

ID	Tipo	N-gramas	Algoritmo	Kernel	Valor C	Reducción de dimensionalidad	Categoría gramatical	Accuracy
69	Binario	1	MultinomialNB	NA	NA	No	Todas	0.78672986
23	Binario	1	SVM	Rbf	10	No	Todas	0.78317536
25	COUNT	1	SVM	Rbf	100	No	Todas	0.76421801
24	TFIDF	1	SVM	Rbf	10	No	Todas	0.76066351
71	COUNT	1	MultinomialNB	NA	NA	No	Todas	0.757109
70	TFIDF	1	MultinomialNB	NA	NA	No	Todas	0.74881517
93	Binario	1	BernoulliNB	NA	NA	No	Todas	0.74763033
94	TFIDF	1	BernoulliNB	NA	NA	No	Todas	0.74763033
95	COUNT	1	BernoulliNB	NA	NA	No	Todas	0.74763033
19	Binario	1	SVM	Rbf	100	No	Verbos todos	0.73933649

Tabla 3-9 Experimentos con categoría gramatical

Los vectores binarios de palabras también obtuvieron buenos resultados usando el algoritmo SVM, como puede verse en la Tabla *3-10*. De hecho, reducir la dimensionalidad a 200 columnas resulta buena estrategia para clasificar. Véase por ejemplo el experimento 56 donde se usan unigrama,s bigramas y trigramas juntos. La reducción a 200 dimensiones ayuda a obtener resultados competitivos. De hecho, se obtiene la misma medida que con unigramas y reducción a 300 dimensiones.

Para el caso de los parámetros de la SVM, el kernel rbf brindó mejor performance con valor C de 10. Por los resultados, es posible determinar que no es buena estrategia eliminar las stop words. Por otra parte, sí resulta buena idea eliminar acentos y truncar las palabras, al menos los mejores tres resultados de la tabla lo muestran. Es importante hacer notar que el mejor resultado de esta tabla supera al mejor resultado de la tabla anterior.

ID	N-gramas	Quitar SW	Quitar acentos	Truncar	Reducción de dimensionalidad	Valor C	Log2	Kernel	Accuracy
11	1	No	Sí	Sí	200	10	None	rbf	0.797393365
12	1	No	Sí	Sí	300	10	None	rbf	0.793838863
56	1,2,3	No	Sí	Sí	200	10	None	rbf	0.793838863
44	1	No	No	Sí	200	10	None	rbf	0.786729858
41	1	No	Sí	No	200	1000	None	rbf	0.783175355
47	1	No	No	No	200	100	None	rbf	0.777251185
59	1	Sí	Sí	Sí	200	1000	None	rbf	0.731042654
50	2	No	Sí	Sí	200	1000	None	rbf	0.704976303
53	3	No	Sí	Sí	200	1000	None	rbf	0.582938389

Tabla 3-10 Vector binario con algoritmo SVM

La medida de *accuracy*, aunque poco, aumenta con vectores de frecuencias de palabras (conteo), esto lo podemos ver en la Tabla 3-11. En la misma tabla podemos observar que nuevamente es el algoritmo SVM con kernel rbf y valor C de 10 el que obtiene mejores resultados, así como reducir las dimensiones de la matriz (200 y 300). Otro patrón que se repite es el de no eliminar stop words y sí eliminar acentos, así como truncar las palabras. El uso de unigramas también es conveniente.

ID	N-gramas	Quitar SW	Quitar acentos	Truncar	Reducción de dimensionalidad	Valor C	Log2	Kernel	Accuracy
8	1	No	Sí	Sí	200	10	None	rbf	0.800947867
4	1	No	Sí	Sí	300	10	None	rbf	0.797393365
13	1	No	Sí	Sí	No	100	None	rbf	0.791469194
54	1,2,3	No	Sí	Sí	200	10	None	rbf	0.79028436
39	1	No	Sí	No	200	10	None	rbf	0.787914692
42	1	No	No	Sí	200	10	None	rbf	0.786729858
7	1	No	Sí	Sí	200	1	None	linear	0.785545024
45	1	No	No	No	200	10	None	rbf	0.780805687
3	1	No	Sí	Sí	300	1	None	linear	0.771327014
57	1	SI	Sí	SI	200	100	None	rbf	0.748815166

Tabla 3-11 Vector conteo con algoritmo SVM

Por otro lado, al utilizar vectores con valores TF-IDF con algoritmo SVM podemos notar que la medida *accuracy* más alta es muy similar a los resultados de la tabla anterior, como muestra la Tabla 3-12. Además, existe un patrón para lograr obtener valores altos para a*ccuracy* ya que

podemos observar una vez más que los resultados más altos se obtienen utilizando el kernel rbf, valor C de 10, sin eliminar stop words, quitando acentos (y no quitando) y truncando. Respecto a la reducción de dimensionalidad, esta vez los resultados más altos fueron con 300 columnas. Como podemos ver en los experimentos 10 y 43, la única diferencia está en quitar los acentos, sin embargo el resultado para la medida *accuracy* es el mismo.

ID	N-gramas	Quitar SW	Quitar acentos	Truncar	Reducción de dimensionalidad	Valor C	Log2	Kernel	Accuracy
10	1	No	Sí	Sí	300	10	None	rbf	0.802132701
43	1	No	No	Sí	300	10	None	Rbf	0.802132701
9	1	No	Sí	Sí	200	10	None	Rbf	0.800947867
40	1	No	Sí	No	300	10	None	Rbf	0.79028436
46	1	No	No	No	300	10	None	Rbf	0.787914692
55	1-3	No	Sí	Sí	300	10	None	Rbf	0.781990521
14	1	No	Sí	Sí	No	10	None	Rbf	0.778436019
58	1	SI	Sí	SI	300	100	None	Rbf	0.755924171
49	2	No	Sí	Sí	300	10	None	rbf	0.708530806
52	3	No	Sí	Sí	300	100	None	Rbf	0.612559242

Tabla 3-12 Vector TF-IDF con algoritmo SVM

Cuando utilizamos el algoritmo Naïve Bayes (NB) podemos notar que la medida *accuracy* es más baja en comparación con los resultados obtenidos con el algoritmo SVM. Los resultados más altos los obtuvimos con el algoritmo NB Multinomial. A pesar de la diferencia de los resultados, una vez más los mejores fueron obtenidos sin eliminar las stop words, quitando acentos y truncando palabras tal y como se muestra en la Tabla 3-13[12].

[12] Esta tabla solo incluye los resultados más altos. Otros experimentos con resultados más bajos no son mostrados.

ID	Tipo	N-gramas	Algoritmo	Quitar SW	Quitar acentos	Truncar	Red Dim	Accuracy
62	Binario	1	MultinomialNB	No	Sí	Sí	NO	0.780805687
60	Conteo	1	MultinomialNB	No	Sí	Sí	NO	0.776066351
84	Conteo	1	BernoulliNB	No	Sí	Sí	NO	0.760663507
85	TFIDF	1	BernoulliNB	No	Sí	Sí	NO	0.760663507
86	Binario	1	BernoulliNB	No	Sí	Sí	NO	0.760663507
61	TFIDF	1	MultinomialNB	No	Sí	Sí	NO	0.747630332
74	Binario	1	GaussianNB	No	Sí	Sí	NO	0.659952607
72	Conteo	1	GaussianNB	No	Sí	Sí	NO	0.657582938
73	TFIDF	1	GaussianNB	No	Sí	Sí	NO	0.654028436

Tabla 3-13 Experimentos con algoritmo Naive Bayes

Hasta este momento, como muestran los resultados, el algoritmo que mejor clasifica anécdotas es una Máquina de Vector Soporte (kernel rbf, valor C de 10) con reducción de dimensionalidad (300) usando unigramas de palabras truncadas, sin acentos y sin eliminar palabras funcionales.

Con el objetivo de comparar los resultados de un método automático con un método basado en conocimiento de una persona, además de los establecidos se propusieron varios baselines basados en presencia de ciertos tiempos verbales (Tabla 3-15). Como ya se dijo antes, la intuición es que las anécdotas pueden caracterizarse por la presencia de verbos en pasado. Esto permitiría pensar que basta con encontrar los comentarios que contengan más verbos en cierto tiempo verbal (por ejemplo, copretérito) para descubrir anécdotas. Si lo anterior fuese cierto, no sería necesario el desarrollo de un método de clasificación automática.

Entonces, se definieron varios cálculos para obtener el promedio de verbos en determinado tiempo verbal. Se usaron dos denominadores para obtener este promedio: la cantidad de verbos en el comentario y la cantidad de palabras en el mismo. Los tiempos y modos verbales utilizados se presentan en la Tabla 3-14 junto con sus combinaciones.

Modo	Tiempo	Ejemplo
Indicativo	pasado simple	dijo, contó
Indicativo	Copretérito	decía, contaba
Subjuntivo	Copretérito	dijera/dijese, contara/contase

Tabla 3-14 Modos y tiempos verbales

La Tabla 3-15 presenta la lista de baselines y el valor de *accuracy* obtenido por cada uno de ellos.

ID	Tipo	Accuracy
B1	V Pas Simple y Copretérito / Tot V > Media = SÍ	0.674170616
B2	V IndSub Pas Simple y Copretérito / Tot V > Media = SÍ	0.665876777
B3	V Copretérito / Tot V > Media = SÍ	0.665876777
B4	V IndSub Copretérito / Tot V > Media = SÍ	0.654028436
B5	V Pas Simple / Tot V > Media = SÍ	0.617298578
B6	Aleatorio respetando distribución entrenamiento	0.509478673
B7	V Copretérito / Tot Pal > Media = SÍ	0.501184834
B8	V IndSub Copretérito / Tot Pal > Media = SÍ	0.501184834
B9	V Pas Simple y Copretérito / Tot Pal > Media = SÍ	0.498815166
B10	V IndSub Pas Simple y Copretérito / Tot Pal > Media = SÍ	0.498815166
B11	V Pas Simple / Tot Pal > Media = SÍ	0.496445498
B12	Aleatorio uniforme	0.478672986

Tabla 3-15 Baselines

3.6.3. Discusión

En esta sección discutiremos los resultados obtenidos para determinar la mejor propuesta de solución al problema de clasificación planteado.

Utilizando la medida de evaluación *accuracy* se encontraron dos mejores clasificadores. Ambos usan un algoritmo de clasificación SVM con kernel rbf y valor C de 10, la medida de *accuracy* de ambos fue 0.8021 (Tabla 3-16). Estos clasificadores resultaron mejores utilizando valores TF-IDF para la representación vectorial de los comentarios, esto es, en lugar de utilizar frecuencias o valores binarios (presencia/ausencia) de palabras resultó mejor darle un peso a cada frecuencia con la medida de TF-IDF. Las características textuales que dieron un mejor resultado fueron los unigramas de palabras, incluyendo *stop words* y con truncamiento a seis caracteres. Lo único que diferenció a los mejores clasificadores por *accuracy* fue que en uno se eliminaron los acentos y en el otro se mantuvieron, esto indica que los acentos no son una característica importante para la clasificación (da los mismo dejarlos que quitarlos).

Por los resultados obtenidos (véase Tabla 3-16), reducir la dimensionalidad del espacio vectorial utilizado a 300 dimensiones es recomendable. Como se mencionó, el método utilizado para reducir la dimensionalidad fue la descomposición de valor singular (SVD). Resulta interesante un resultado con reducción SVD, ya que esta ha sido utilizada para análisis semántico de documentos mediante técnicas como el análisis de semántica latente (Latent Semantic

Analysis, LSA) o el indexado de semántica latente (Latent Semantic Indexing, LSI) (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990). Lo anterior puede indicar que la clasificación mejora al utilizar características internas o subyacentes de la colección de datos.

ID	Tipo	N-gramas	Quitar SW	Quitar acentos	Truncar	Red Dim	Log2	Algoritmo	Kernel	Valor C
10	TFIDF	1	No	Sí	Sí	300	No	SVM	rbf	10
43	TFIDF	1	No	No	Sí	300	No	SVM	rbf	10

ID	Accuracy	Precision	Recall	F-score
10	0.8021	0.8277	0.7630	0.7940
43	0.8021	0.8328	0.7559	0.7925

Tabla 3-16 Mejores clasificadores por *accuracy*

Según el reporte de evaluación del clasificador con ID 10, que se muestra enseguida, se clasifican como anécdotas 322 de las 422 y como no anécdotas 355 de las 422 reales (véase Tabla 3-17). El clasificador toma como anécdotas 67 que no son anécdotas (FP) y clasifica como no anécdotas 100 que sí lo son (FN). Por lo tanto podemos decir que este clasificador pierde 100 anécdotas que sí lo son.

```
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.8021327014218009
Accuracy: 0.8021327014218009
Precision: 0.8277634961439588
Recall: 0.7630331753554502
F-score: 0.7940813810110975
Matriz de confusión:
[[322 100]
 [ 67 355]]
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.83      0.76      0.79       422
         NO       0.78      0.84      0.81       422

avg / total       0.80      0.80      0.80       844
```

		Predichas		
		Anécdota	No anécdota	Total
Reales	Anécdota	322	100	422
	No anécdota	67	355	422
	Total	389	455	

Tabla 3-17 Matriz de confusión mejor clasificador por *accuracy* (ID 10)

En cuanto al segundo mejor clasificador por *accuracy* (véase Tabla 3-18), el reporte de evaluación muestra que detecta 3 anécdotas menos (322 vs 319); sin embargo, tiene 3 menos falsos positivos (67 vs 64) lo que le da una medida de *precision* un poco más alta que la del otro clasificador (0.8277 vs 0.8328).

```
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.8021327014218009
Accuracy: 0.8021327014218009
Precision: 0.8328981723237598
Recall: 0.7559241706161137
F-score: 0.7925465838509317
Matriz de confusión:
[[319 103]
 [ 64 358]]
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.83      0.76      0.79       422
         NO       0.78      0.85      0.81       422

avg / total       0.80      0.80      0.80       844
```

		Predichas		
		Anécdota	No anécdota	Total
Reales	Anécdota	319	103	422
	No anécdota	64	358	422
	Total	383	461	

Tabla 3-18 Matriz de confusión mejor clasificador por *accuracy* (ID 43)

La medida de *precision* nos muestra el comportamiento del clasificador con respecto a los falsos positivos de la clase anécdota. En otras palabras, evalúa de las anécdotas predichas por el clasificador cuántas realmente lo son. Por lo tanto podríamos intentar seleccionar el clasificador con mayor medida de *precision*. Si fuera así, el mejor clasificador sería uno basado en frecuencia

de unigramas de palabras, incluyendo palabras funcionales, sin acento y truncadas a seis caracteres como se muestra en la Tabla 3-19. En este caso el algoritmo de clasificación sería un Naïve Bayes con distribución de Bernoulli, que asume que cada característica tiene un valor binario.

ID	Tipo	N-gramas	Quitar SW	Quitar acentos	Truncar	Red Dim	Log2	Algoritmo	Kernel	Valor C
84	Conteo	1	No	Sí	Sí	No	No	BernoulliNB	NA	NA

ID	Accuracy	Precision	Recall	F-score
84	0.7606	0.8416	0.6421	0.7284

Tabla 3-19 Mejor clasificador por *precision*

Ahora bien, si observamos el informe de este clasificador, el cual se muestra enseguida, este clasificador ciertamente se equivoca menos el detectar anécdotas, solo 51 no anécdotas son clasificadas como tales (Tabla 3-20). Sin embargo también se puede observar que el total de anécdotas correctamente detectadas (271) es mucho menor que las obtenidas por el mejor clasificador por *accuracy* (322). También se puede ver que clasifica como anécdotas más comentarios que no lo son (151). Es claro que este clasificador se equivoca menos, pero recupera menos anécdotas y tiene más falsos positivos.

```
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.7606635071090048
Accuracy: 0.7606635071090048
Precision: 0.8416149068322981
Recall: 0.6421800947867299
F-score: 0.728494623655914
Matriz de confusión:
[[271 151]
 [ 51 371]]
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.84      0.64      0.73       422
         NO       0.71      0.88      0.79       422

avg / total       0.78      0.76      0.76       844
```

		Predichas		
		Anécdota	No anécdota	Total
Reales	Anécdota	271	151	422
	No anécdota	51	371	422
	Total	322	522	

Tabla 3-20 Matriz de confusión mejor clasificador por *precision*

La medida *recall* nos indica el comportamiento del clasificador con respecto a los FN. Esto es, nos indica de las anécdotas reales, cuántas fueron clasificadas correctamente. Si tomamos el valor con el *recall* más alto, que es 0.9526, nos encontraremos con un modelo generado a partir de etiquetas POS utilizando todos los verbos, con un vector binario y el algoritmo GaussianNB (véase Tabla 3-21).

ID	Tipo	N-gramas	Quitar SW	Quitar acentos	Truncar	Red Dim	Log2	Algoritmo	Kernel	Valor C
77	Binario	NA	NA	NA	NA	No	No	GaussianNB	NA	NA

ID	Accuracy	Precision	Recall	F-score
77	0.5284	0.5153	0.9526	0.6688

Tabla 3-21 Mejor clasificador por *recall*

En el reporte generado que se muestra a continuación podemos observar que el modelo asigno correctamente 402 anécdotas (TP) de un total de 422, sin embargo podemos observar que el modelo asigno 378 veces la clase anécdota cuando la clase real era No anécdota (FP) (véase Tabla 3-22).

```
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.5284360189573459
--- Medidas Scikit Learn ---
Accuracy: 0.5284360189573459
Precision: 0.5153846153846153
Recall: 0.95260663507109
F-score: 0.6688851913477537
Matriz de confusión:
[[402  20]
 [378  44]]
```

```
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.52      0.95      0.67       422
         NO       0.69      0.10      0.18       422

avg / total       0.60      0.53      0.42       844
```

		Predichas		
		Anécdota	No anécdota	Total
Reales	Anécdota	402	20	422
	No anécdota	378	44	422
	Total	780	64	

Tabla 3-22 Matriz de confusión mejor clasificador por *recall*

Ahora bien, si tomamos en cuenta la medida F-score, como nos muestra la Tabla 3-23, podemos encontrar que el modelo clasificador con la medida más alta (0.7991) corresponde a un modelo generado con un vectorizador de conteo y un algoritmo NB que es MultinomialNB, usando stop words y quitando acentos, truncando palabras y sin reducir la dimensionalidad.

ID	Tipo	N-gramas	Quitar SW	Quitar acentos	Truncar	Red Dim	Log2	Algoritmo	Kernel	Valor C
60	Conteo	1	NO	SI	SI	No	None	MultinomialNB	NA	NA

ID	Accuracy	Precision	Recall	F-score
60	0.7760	0.7244	0.8909	0.7991

Tabla 3-23 Mejor clasificador por F-score

```
********** EVALUACIÓN **********
Medida de clasificación Accuracy: 0.7760663507109005
Accuracy: 0.7760663507109005
Precision: 0.7244701348747592
Recall: 0.8909952606635071
F-score: 0.7991498405951116
Matriz de confusión:
[[376  46]
 [143 279]]
Reporte de clasificación:
             precision    recall  f1-score   support

         SI       0.72      0.89      0.80       422
         NO       0.86      0.66      0.75       422
```

```
avg / total       0.79      0.78      0.77       844
```

		Predichas		
		Anécdota	No anécdota	Total
Reales	Anécdota	376	46	422
	No anécdota	143	279	422
	Total	519	325	

Tabla 3-24 Matriz de confusión mejor clasificador por F-score

Si observamos la matriz de confusión del mejor resultado por *F-score* (véase Tabla 3-24), podemos observar que este clasificador recupera más anécdotas que el mejor clasificador por *accuracy*. Sin embargo, aunque no obtiene tantos FP como el mejor clasificador por *recall*, sí obtiene más que el mejor clasificador por *accuracy.* En otras palabras, este clasificador es mejor para detectar anécdotas, pero no es tan bueno para detectar no anécdotas.

En conclusión podemos decir que hay dos clasificadores que podrían ser usados en una aplicación real. Uno recupera más anécdotas verdaderas (TP=376), pero con más anécdotas que no lo son (FP=143), este es el mejor clasificador por *F*-score. Mientras que el otro recupera menos anécdotas verdaderas que el primer clasificador (TP=322), pero recupera menos anécdotas que no lo son (FP=67), este es el mejor clasificador por *accuracy*.

La selección del clasificador depende, entre otras cosas, del tipo de objeto o evento que se quiere detectar. Por ejemplo, en el caso de fraudes de tarjetas de crédito podría preferirse el clasificador que recupera muchos fraudes (TP), aunque también recupere muchos que no lo son (FP). En nuestro caso decidimos usar el mejor clasificador por *accuracy* para el caso de una aplicación real, pensando en que sea mejor la detección de anécdotas, aunque no se detecten todas.

4. Recuperación automática de fotografías

En este capítulo se explica la tarea de recuperación automática de fotografías a través de descriptores. El objetivo principal es asociar estos descriptores (palabras clave) con fotografías, ya que como habíamos mencionado anteriormente, debido a la gran cantidad de fotografías que están publicadas en la página de Facebook *La ciudad de México en el tiempo*, resulta difícil la búsqueda de una imagen específica.

Al principio de esta tesis planteamos como objetivo desarrollar un método para la recuperación automática de fotografías con el fin de automatizar la búsqueda de imágenes de la página realizando esta tarea de una forma más rápida. En los siguientes apartados se describe el proceso de obtención y evaluación de descriptores para fotografías con el fin de lograr este objetivo.

Para realizar la recuperación automática de las fotografías nos basamos en una técnica de la recuperación de información que es la búsqueda por palabras clave (Gelbukh & Sidorov, 2010). Para cada una de las fotografías que se encuentran en la página hemos asignando diferentes descriptores o palabras clave. Estas palabras las obtuvimos con base en la medida TF-IFD que, como ya hemos mencionado, es una medida que se encarga de establecer la relevancia de cada palabra de un documento que se encuentra en una colección. Dicha medida fue obtenida para los comentarios de cada fotografía, lo que quiere decir que las palabras clave o descriptores de la fotografía fueron obtenidos de los comentarios que la gente publica en el sitio.

4.1.Recopilación del corpus

El primer paso fue la recopilación del corpus de comentarios, es decir, la colección de documentos. En este caso, cada uno de los documentos está conformado por todos los comentarios de una fotografía. Como las fotografías están organizadas en diferentes álbumes, consideramos a cada uno de los álbumes como una colección de documentos, y de esta forma obtuvimos la medida TF-IDF por cada colección[13].

[13] Estamos conscientes de que también se pudo calcular la medida TF-IDF usando todos los álbumes al mismo tiempo, esto es, como una sola colección. Sin embargo, decidimos adoptar la estrategia de considerar cada álbum como una colección como primer acercamiento para resolver el problema. En un futuro se podrán hacer experimentos con la otra estrategia.

Para obtener el corpus también utilizamos la herramienta FQL de Facebook que mencionamos en el apartado 3.1. Como recordaremos, esta herramienta nos permite realizar consultas a la base de datos de Facebook y es similar al lenguaje SQL. También utilizamos el lenguaje de programación Python con la librería urllib2 para realizar peticiones HTTP. A continuación se describe el proceso de recopilación del corpus de comentarios. Este proceso fue realizado para cada álbum de la página.

Primero se obtuvo el *objet_id* (id de un objeto en Facebook) de cada una de las imágenes con la siguiente instrucción donde `aid` es el id del álbum.

```
http://graph.facebook.com/fql?q=select object_id from photo
where aid="187533597935335_43100"
```

Como recordaremos, la salida de las consultas es en formato Json y realizando esta consulta pudimos consultar el object_id de cada imagen del álbum. Tomando en cuenta la salida anterior realizamos la siguiente consulta por cada imagen (por cada *object_id*) en donde consultamos, de la tabla *photo*, información general de cada imagen.

```
https://graph.facebook.com/fql?q=select aid, object_id,
caption, src_big from photo where aid="187533597935335_42656"
and object_id = 1034727979882555 limit 8000
```

En la Figura 4-1 se muestra la salida de la consulta anterior en donde se encuentran, entre otros datos, la descripción en el campo *caption* y la url de la imagen en el campo *src_big* que utilizaremos para saber a qué imagen pertenece cada comentario.

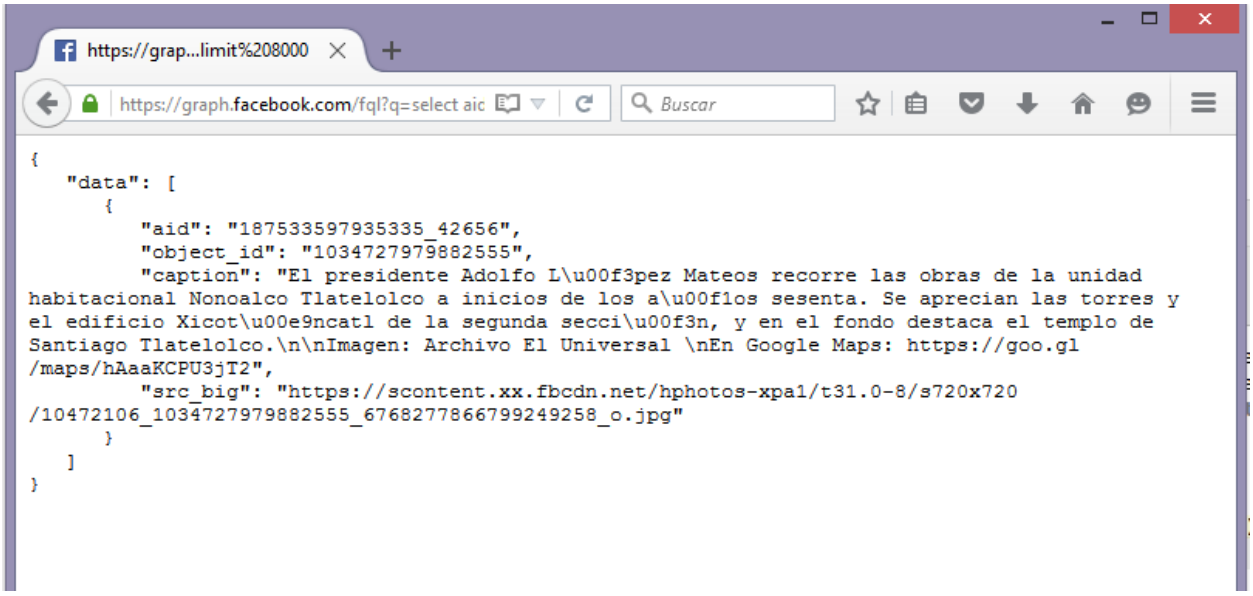


Figura 4-1 Salida de consulta a la tabla *photo*

La siguiente consulta es para obtener los comentarios realizados a una fotografía. Para ello vamos a realizar una consulta a la tabla *comment* donde el *objetc_id* es el id de la imagen.

```
http://graph.facebook.com/fql?q= select object_id, id, text, fromid from comment where object_id=1034727979882555 limit 8000
```

La salida de la consulta anterior se muestra en la Figura 4-2. Cada elemento del formato Json corresponde a un comentario y el campo *text* es donde se muestra el comentario.

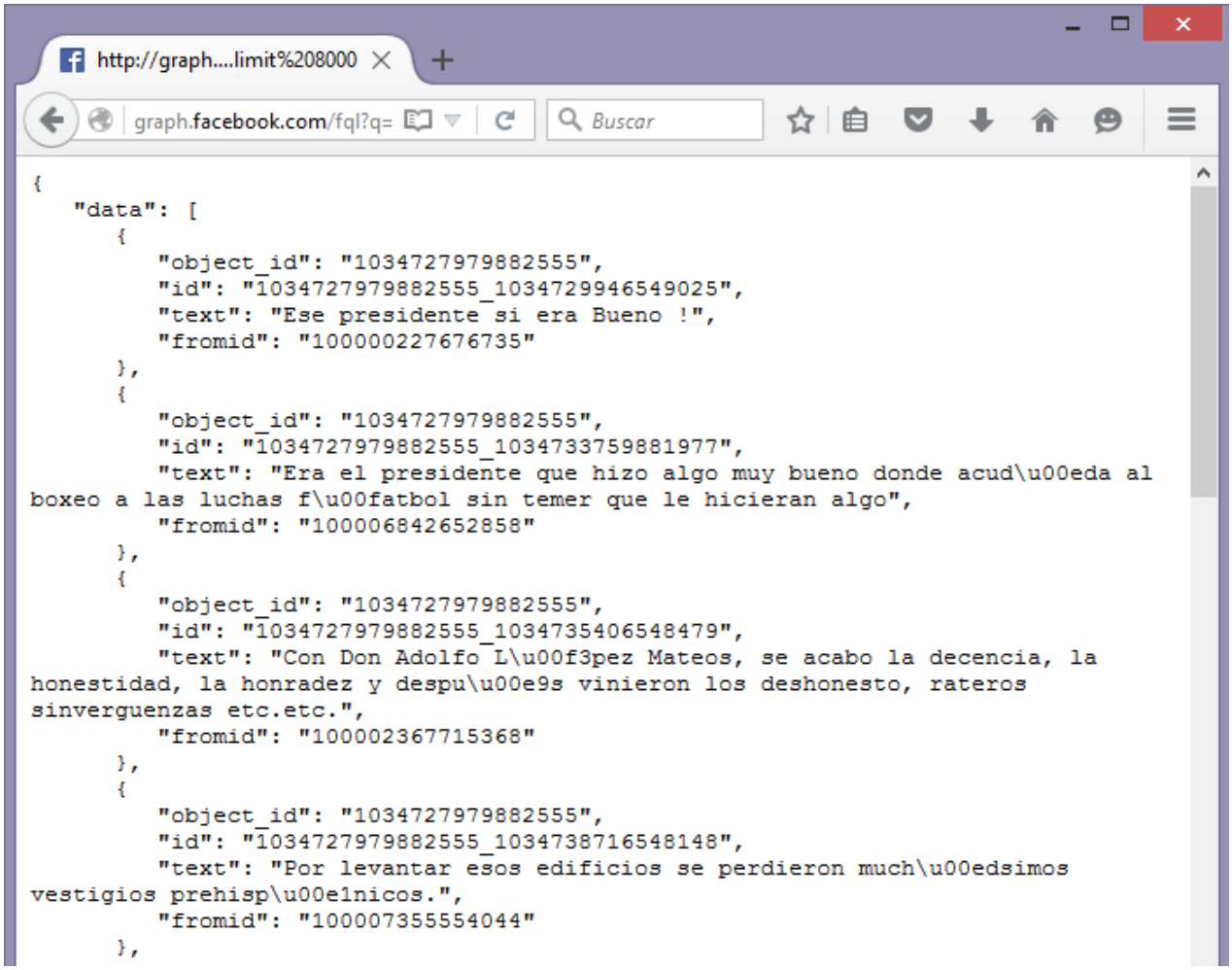


Figura 4-2 Salida de la consulta a la tabla *comment*

Utilizando estas consultas realizamos un programa en Python que permitiera ejecutarlas para todas las fotografías. La salida fue almacenada en un documento de texto que fue nombrado con el *objetc_id* para identificar de qué imagen es la información.

Como estas instrucciones fueron ejecutadas por cada álbum, al final obtuvimos diferentes carpetas (una por cada álbum) que contenían la información de cada una de las imágenes en formato Json. Por cada fotografía se tuvieron dos archivos diferentes, el primero donde se

encuentra la información general de la imagen y el segundo donde se encuentra la información de los comentarios.

4.2.Preprocesamiento

Hasta este punto, tenemos diferentes carpetas que guardan información de cada álbum de fotografías. Los comentarios fueron almacenados en formato Json, así que el siguiente paso consiste en crear documentos donde se encuentren únicamente los comentarios, es decir, el contenido del campo *text*. Para lograr esto, realizamos un programa en Python donde la entrada son los archivos Json con la información de cada comentario y la salida son archivos que contienen únicamente texto, todos los comentarios de cada fotografía forman un documento.

En este paso de preprocesamiento, los acentos fueron eliminados y las palabras fueron pasadas a minúsculas ya que para la obtención de descriptores vamos a tomar en cuenta la frecuencia de aparición de las palabras y el hecho de usar mayúsculas o acentos puede alterar el resultado. De hecho, consideramos que la manera de escribir en redes sociales no sigue necesariamente las normas de ortografía, por lo que intuimos que será mejor normalizar las palabras quitándoles acentos y mayúsculas.

4.3.Obtención de descriptores de fotografías

Una vez que tenemos las colecciones de documentos, el siguiente paso es la obtención de los descriptores (palabras clave). La forma de realizarlo fue obteniendo la medida TF-IDF de cada palabra[14]. Para lograrlo utilizamos la librería *Scikit-learn* para Python y el proceso fue repetido para cada álbum de fotografías. Se decidió usar esta medida porque es un estándar en tareas de recuperación de información, aunque existen otros métodos para obtener palabras clave que podrán usarse en investigaciones futuras. Es pertinente recordar que esta medida asigna mayor peso a palabras de alta frecuencia que ocurren en pocos documentos y peso bajo a palabras que están en muchos documentos de la colección.

Primero utilizamos la función *TfidfVectorizer*[15] que se encarga de convertir una colección de documentos en una matriz de características TF-IDF. Lo primero que realiza el programa es cargar todos los documentos de una colección, es decir, los documentos de un álbum de

[14] El proceso de obtención de la medida TF-IDF se explicó en el apartado 2.3.3

[15] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

fotografías, después esta función se encarga de crear una matriz en donde el encabezado está conformado por las diferentes palabras (tipos) de todos los documentos (vocabulario) excepto las *stop words,* que como recordaremos son artículos, pronombres, preposiciones, etc. Las *stop words* son palabras que por sí solas no tienen un significado y, como buscamos palabras que describan una imagen, consideramos que estas palabras no nos ayudarían a lograr el objetivo. Cada renglón de la matriz representa un documento (vector) y sus valores son las medidas TF-IDF asignados a cada palabra.

Hasta este punto tenemos la medida TF-IDF de todas las palabras; sin embargo, no todas las palabras describen a la imagen. Es necesario establecer un umbral de la medida TF-IDF para determinar qué palabras serán descriptores. Para saber qué palabras tienen más relevancia en los documentos, obtuvimos el promedio y la desviación estándar de la medida TF-IDF de todas las palabras de cada documento. Decidimos experimentar con la suma del promedio y una desviación estándar como umbral para saber qué palabras tomaremos en cuenta como descriptores de cada imagen. En otras palabras, si el valor TF-IDF de una palabra es mayor a la suma del promedio más una desviación estándar, entonces la palabra es considerada como descriptor de la imagen. Finalmente estas palabras fueron almacenadas en diferentes documentos de texto (uno por cada imagen) los cuales nombramos con el *Facebook ID* para identificar la imagen a la que corresponden.

Figura 4-3 Fotografía del álbum *La ciudad Universitaria*

Como ejemplo de este proceso tomaremos la imagen que se muestra en la Figura 4-3 que pertenece al álbum *La ciudad Universitaria.* Primero se obtuvo la media TF-IDF para cada palabra de los comentarios realizados a la imagen, posteriormente se obtuvo el promedio de estos valores que dio como resultado 0.0505 y después la desviación estándar obteniendo 0.0374, la suma de estas cantidades da un resultado de 0.0879. Así, los descriptores para esta imagen son las palabras que se muestran en la Tabla 4-1, ya que tienen un valor mayor a este.

Palabra	Valor TF-IDF
Ofunam	0.3498
Orquesta	0.2794
Bajo	0.2794
Mural	0.2220
Viernes	0.2196
Difusión	0.1643
Facultades	0.1342
Conciertos	0.1252
Concierto	0.1198
Resanacion	0.1095
2011	0.1095
Escuchar	0.0999
Trabajos	0.0931

Tabla 4-1 Ejemplo de descriptores de una imagen

De las palabras mostradas en la Tabla 4-1, se puede observar que la mayoría describe bien a la imagen. Las únicas palabras que tal vez resaltan como poco relacionadas son: resanacion y trabajos. Esta situación deriva del hecho de que se están usando comentarios de una red social, donde los usuarios son libres de tratar cualquier tema, incluso temas que no tengan que ver con la imagen. Piénsese en recuerdos que la imagen pueda evocar en el usuario y que lo lleven a platicar anécdotas que se alejen del contenido de la imagen. Por esta razón propusimos la realización de dos experimentos de obtención de descriptores: uno con todos los comentarios y otro solo de los comentarios clasificados como No anécdotas. La intuición detrás de estos experimentos es que las No anécdotas contendrán información más relacionada con la imagen y por lo tanto permitirán obtener mejores descriptores.

Para saber si la intuición era cierta, hemos utilizado el modelo clasificador generado en la fase de recuperación automática de anécdotas para clasificar los comentarios de las imágenes. Una vez separados los comentarios en Anécdotas y No anécdotas, hemos utilizado las No anécdotas para generar los descriptores de imágenes.

Al final se obtuvieron dos conjuntos de palabras que pueden describir una imagen. Para saber que método puede resultar mejor para generar descriptores de imágenes, si utilizando todos los comentarios o utilizando únicamente los comentarios que pertenecen a la clase No anécdotas, realizamos una evaluación utilizando estos dos conjuntos de palabras. Esta evaluación se presenta en la siguiente sección.

4.4.Evaluación y resultados

Una vez que obtuvimos los descriptores de imágenes es necesario saber que tan acertado es este método para generar estos descriptores, para ello hemos realizado una evaluación. Esta evaluación nos puede dar indicios para saber si verdaderamente las palabras describen a las imágenes y también para saber si existe una diferencia entre usar todos los comentarios y solo las No anécdotas.

En los siguientes apartados se describe el proceso de evaluación de estos descriptores para lo cual nos dimos a la tarea de buscar la colaboración de personas con diferentes perfiles para que nos ayudarán a realizar esta evaluación, finalmente se muestran los resultados del proceso de evaluación y que método funciona mejor para generar descriptores de imágenes a través de comentarios en *Facebook.*

4.4.1. Evaluación

Hasta este punto tenemos los descriptores o palabras clave en documentos con formato txt. En este apartado se describe la forma de evaluar estas palabras en relación a la imagen que describen. Para lograrlo implementamos un buscador de imágenes en una interfaz web que se encarga de la recuperación de fotografías a través de las palabras clave que hemos obtenido de los comentarios. En esta interfaz web, que se muestra en la Figura 4-4, el usuario puede ingresar palabras y buscar imágenes que se relacionen con estas palabras.

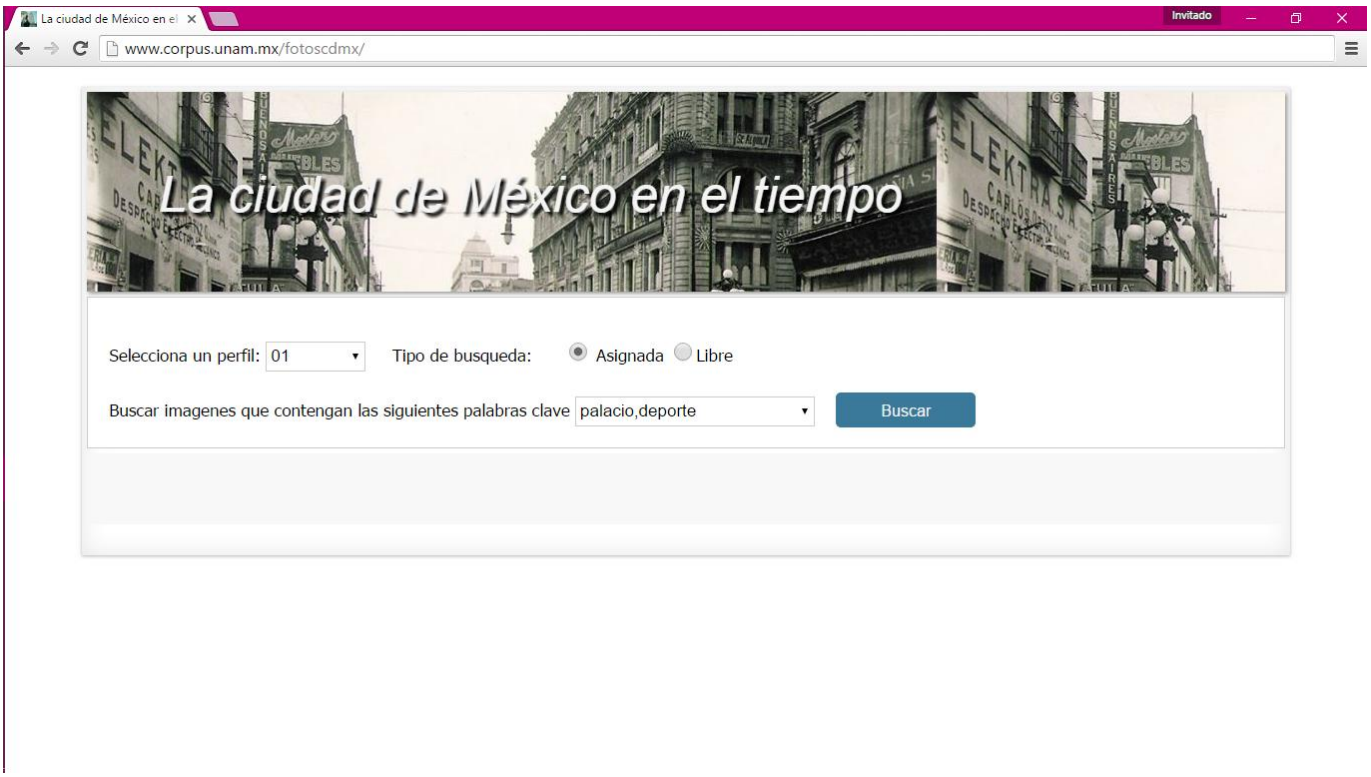


Figura 4-4 Interfaz para el buscador de imágenes

La evaluación se realizó a través del buscador de imágenes, para esto solicitamos la colaboración de 12 personas con diferentes perfiles. Llamamos universitarios a las personas que tuvieron estudios a nivel de universidad y no universitarios a los que no tuvieron.

- 4 jóvenes (universitarios y no universitarios) de 19-29
- 4 adultos (universitarios y no universitarios) de 30-59
- 4 adultos mayores (universitarios y no universitarios) 60 +

Dividimos al grupo de personas de modo que 6 personas evaluaran los descriptores generados con todos los comentarios y 6 personas evaluaran los descriptores generados con los comentarios clasificados como No anécdotas. Como se muestra en la Tabla 4-2 y Tabla 4-3 asignamos un ID a cada persona, que utilizamos para conocer el perfil y el conjunto de palabras que evaluaría.

ID	Perfil
01	Joven universitario
02	Joven no universitario
03	Adulto universitario
04	Adulto no universitario
05	Adulto mayor universitario
06	Adulto mayor no universitario

Tabla 4-2 Perfiles para evaluación de palabras clave generadas con TODOS los comentarios

ID	Perfil
07	Joven universitario
08	Joven no universitario
09	Adulto universitario
10	Adulto no universitario
11	Adulto mayor universitario
12	Adulto mayor no universitario

Tabla 4-3 Perfiles para evaluación de palabras clave generadas solo con los comentarios NO anécdotas

Para evaluar diferentes palabras hemos pedido a las personas que seleccionen el ID de su perfil y el tipo de búsqueda como se muestra en la interfaz del buscador de la Figura 4-4. Los tipos de búsqueda son: asignada y libre. En la primera hemos pedido que realicen la búsqueda de 6 palabras que nosotros establecimos y en la segunda hemos pedido que busquen palabras que ellos decidan, es decir que hagan una búsqueda libre. La intención de estos dos tipos de búsqueda es tratar de evaluar el rendimiento del método en una situación controlada y en una situación no controlada para ver el rendimiento en un entorno real.

Para realizar el tipo de búsqueda asignado hemos establecido las siguientes palabras:

1. Palacio, deporte
2. Estadio, azteca
3. Monumento, revolución
4. Torre, latino
5. zócalo, catedral
6. Palacio, bellas, artes

Al buscar una o más palabras, como se muestra en la Figura 4-5, la salida serán las imágenes que se asocian con las palabras que se ingresaron. Luego, las personas deberán elegir entre las siguientes opciones de acuerdo a si existe o no una relación entre las palabras y la imagen recuperada.

- Si
- No
- No sé

Figura 4-5 Interfaz de evaluación de búsqueda

Para analizar los resultados de la evaluación obtuvimos la siguiente información por cada perfil, de esta forma es posible saber si existe una diferencia al generar las palabras clave de diferente forma (con y sin anécdotas).

- Tipo de búsqueda
- Palabra o palabras buscadas
- Número de imágenes que muestra la pantalla

- Número de imágenes con la opción Si
- Número de imágenes con la opción No
- Número de imágenes con la opción No sé

4.4.2. Resultados

A continuación se presentan los resultados de la evaluación de la recuperación de imágenes (fotografías) por medio de descriptores. Los resultados se presentan en forma de tablas que muestran el total de imágenes evaluadas, el número de imágenes evaluadas con Sí, No y No se respectivamente y el porcentaje de estas respuestas. Los resultados se presentan divididos por el método por el que fueron generados los descriptores (todos los comentarios y solo No anécdotas) ya que de esta forma podemos ver la diferencia entre ambos métodos.

En la Tabla 4-4 se presenta el total de imágenes evaluadas por las personas tomando en cuenta los dos tipos de búsqueda (asignada y libre), esto para tener un panorama general de los dos métodos. Primero, se observa claramente que se presentaron más fotografías (1,846) cuando los descriptores fueron generados con todos los comentarios a diferencia del método donde se los descriptores se generan con los comentarios pertenecientes a la clase No anécdotas (515).

Método	Total de imágenes evaluadas	Sí	No	No sé
Todos	1,846 (100%)	1,152 (62.41%)	391 (21.18%)	303 (16.41%)
No anécdotas	515 (100%)	346 (67.18%)	116 (22.52%)	53 (10.29%)

Tabla 4-4 Resultados para ambos métodos

También podemos observar que el porcentaje de respuestas Si, donde las imágenes presentadas si eran descritas por las palabras clave, es mayor para el método No anécdotas con un 67.18% a diferencia del método donde se utilizaron todos los comentarios que tiene un porcentaje de 62.41%. Otro aspecto a resaltar es la disminución de respuestas No sé con el conjunto de solo No anécdotas (más de seis puntos porcentuales). Estos hechos pueden llevarnos a pensar que el método es mejor cuando se usan solo No anécdotas.

En la Tabla 4-5 los resultados están divididos de acuerdo al tipo de búsqueda (asignada y libre) con el objetivo de conocer la diferencia entre los diferentes datos de entrada. Esto nos da una idea del rendimiento del método considerando datos controlados (búsqueda asignada), ya que nosotros conocíamos las imágenes que se presentaban al buscar las palabras asignadas, y

datos no controlados (búsqueda libre), en donde la persona puede buscar cualquier palabra y nosotros no conocíamos las imágenes que el sistema presentaría.

Método	Tipo de búsqueda	Total de imágenes evaluadas	Sí	No	No sé
Todos	Asignada	823 (100%)	567 (68.89%)	178 (21.63%)	78 (9.48%)
	Libre	1,023 (100%)	585 (57.18%)	213 (20.82%)	225 (21.99%)
No anécdota	Asignada	376 (100%)	260 (69.15%)	80 (21.28%)	36 (9.57%)
	Libre	139 (100%)	86 (61.87%)	36 (25.90%)	17 (12.23%)

Tabla 4-5 Resultados por tipo de búsqueda

El mayor porcentaje de respuestas Sí es de 69.15%, que pertenece al tipo de búsqueda Asignada y a los descriptores generados únicamente con comentarios que son No anécdotas. Por otro lado, el porcentaje más bajo de respuestas Sí es de 57.18% que corresponde a la búsqueda libre del método que utiliza todos los comentarios. Es claro que el sistema disminuye su rendimiento en la búsqueda libre (de 68.89% a 57.18%, -11.71%), especialmente para el método con todos los comentarios donde se ve un fuerte incremento en la respuesta No sé. En el caso del método con no anécdotas, aunque existe también una disminución del rendimiento, esta no es tan pronunciada (de 69.15% a 61.87%, -7.28)

Algo interesante es que los porcentajes de respuesta No sé disminuyen de forma importante para el conjunto de No anécdotas en búsqueda libre y el porcentaje de respuesta Sí aumenta en los dos tipos de búsqueda para las No anécdotas (aunque muy poco para la búsqueda asignada). En la búsqueda libre, el método de No anécdotas disminuye las imágenes que causan duda (respuesta No sé: 21.99% > 12.23%) y aumenta tanto las imágenes no relacionadas (respuesta No: 20.82% < 25.90%) y como las relacionadas (respuesta Sí: 57.18% < 61.87%). Lo anterior vuelve a dar indicios de que el método con No anécdotas tiene mejor rendimiento.

También tomamos en cuenta diferentes perfiles de edad para evaluar las fotografías de la Ciudad de México. La Tabla 4-6 muestra los resultados tomando en cuenta el rango de edad de las personas que nos ayudaron a evaluar las fotografías.

Método	Edad	Total de imágenes evaluadas	Sí	No	No sé
Todos	Joven	751 (100%)	445 (59.25%)	167 (22.24%)	139 (18.51%)
	Adulto	566 (100%)	345 (60.95%)	134 (23.67%)	87 (15.37%)
	Adulto mayor	529 (100%)	362 (68.43%)	90 (17.01%)	77 (14.56%)
No anécdota	Joven	204 (100%)	153 (75%)	21 (10.29%)	
	Adulto	139 (100%)	89 (64.03%)	41 (29.50%)	9 (6.47%)
	Adulto mayor	172 (100%)	104 (60.47%)	54 (31.40%)	14 (8.14%)

Tabla 4-6 Resultados por edad

Para el método de todos los comentarios el mayor porcentaje de respuestas Sí fue de 68.43% que corresponde al perfil de edad Adulto Mayor (60+), el porcentaje más bajo de respuestas No también fue del perfil Adulto Mayor con 17.01%. Para el método con no anécdotas el mayor porcentaje fue de 75% del perfil de edad Joven (19-29 años) que también tiene el porcentaje más bajo con respuestas No, que fue de 10.29%. Hasta ahora los porcentajes más altos son los que resultaron de la evaluación de método que genera los descriptores con los comentarios que no son anécdotas. Por lo que se observa en esta tabla, parece que el sistema mejora su rendimiento para jóvenes cuando se usan solo las no anécdotas, pero empeora para los adultos mayores y en el caso del método con todos los comentarios ocurre lo contrario, aumenta para adulto mayor y disminuye para jóvenes. Como se dijo antes, el método con todos los comentarios arroja considerablemente más imágenes y que parezca que tiene mejor rendimiento en adultos puede ser consecuencia de que los adultos reconozcan más imágenes antiguas o históricas, situación que se provoca por el tipo de sitio Web sobre el que se hizo la investigación. Sin embargo, que el método con solo no anécdotas tenga mejor rendimiento para jóvenes llama la atención porque parecería que de alguna manera se filtraron imágenes más antiguas.

Otra característica que tomamos en cuenta en las personas que nos ayudaron a evaluar fue su nivel de estudio (universitario, no universitario), en la Tabla 4-7 se presentan los resultados divididos según el nivel de estudio de los participantes en la evaluación. Podemos observar que para los universitarios, el mayor porcentaje de respuestas Sí se da con todos los comentarios (69.28%) y disminuye cuando se usan solo no anécdotas (66.04%). Para los no universitarios sucede lo contrario, esto es, el porcentaje de respuestas Sí es mayor con las no anécdotas (69.07%) que con todos los comentarios (55.79%). Nuevamente parece que el sistema tiene rendimiento diferente para distintos grupos de personas. Aquí, el sistema parece tener mejor rendimiento con todos los comentarios para los universitarios, y mejor rendimiento con solo las

no anécdotas para los no universitarios. En este caso no es claro qué tipo de imágenes estaría filtrando el método de No anécdota.

Método	Nivel de estudio	Total de imágenes evaluadas	Sí	No	No sé
Todos	Universitario	905 (100%)	627 (69.28%)	121 (13.37%)	157 (17.35%)
	No universitario	941 (100%)	525 (55.79%)	270 (28.69%)	146 (15.52%)
No anécdota	Universitario	321 (100%)	212 (66.04%)	84 (26.17%)	25 (7.79%)
	No universitario	194 (100%)	134 (69.07%)	32 (16.49%)	28 (14.43%)

Tabla 4-7 Resultados por nivel de estudio

En la Tabla 4-8 buscamos comparar los resultados tomando en cuenta el tipo de búsqueda (asignada y libre) y la edad de los participantes (joven, adulto, adulto mayor) para ambos métodos. Primero vemos en general que el porcentaje más alto de respuestas Sí es de 77.27% que corresponde una vez más al método con No anécdotas, en este caso con el perfil de edad Joven y búsqueda asignada. Para el método que incluye todos los comentarios el mayor porcentaje para respuesta Sí es de 71.27% del perfil Adulto mayor en búsqueda libre.

Método	Tipo de búsqueda	Edad	Total de imágenes evaluadas	Sí	No	No sé
Todos	Asignada	Joven	273 (100%)	183 (67.03%)	69 (25.27%)	21 (7.69%)
		Adulto	275 (100%)	188 (68.36%)	63 (22.91%)	24 (8.73%)
		Adulto mayor	275 (100%)	196 (71.27%)	46 (16.73%)	33 (12%)
	Libre	Joven	478 (100%)	262 (54.81%)	98 (20.50%)	118 (24.69%)
		Adulto	291 (100%)	157 (53.95%)	71 (24.40%)	63 (21.65%)
		Adulto mayor	254 (100%)	166 (65.35%)	44 (17.32%)	44 (17.32%)
No anécdota	Asignada	Joven	154 (100%)	119 (77.27%)	17 (11.04%)	18 (11.69%)
		Adulto	63 (100%)	46 (73.02%)	12 (10.05%)	5 (7.94%)
		Adulto mayor	159 (100%)	95 (59.75%)	51 (32.08%)	13 (8.18%)
	Libre	Joven	50 (100%)	34 (68%)	4 (8%)	12 (24%)
		Adulto	76 (100%)	43 (56.58%)	29 (38.16%)	4 (5.26%)
		Adulto mayor	13 (100%)	9 (69.23%)	3 (23.08%)	1 (7.69%)

Tabla 4-8 Resultados por tipo de búsqueda y edad

En este análisis más detallado, podemos volver a observar que el rendimiento del sistema aumenta para el método No anécdota en jóvenes, pero solo en búsqueda asignada (77.27). En la búsqueda

libre siguen teniendo mejor rendimiento para adultos mayores (69.23%). Una tendencia que podemos confirmar es que el sistema tiene mejor rendimiento con No anécdota en todos los perfiles, excepto en adulto mayor con búsqueda asignada, donde baja drásticamente de 71.27% a 59.75% y aumentan las respuestas No de 16.73% a 32.08%.

También comparamos los resultados del tipo de búsqueda y el nivel de estudio en ambos métodos. La Tabla 4-9 muestra estos resultados donde se toman en cuenta estas dos característica. El porcentaje más alto de respuesta Sí es de 75%, en el tipo de búsqueda asignada, el nivel de estudio universitario y esta vez el porcentaje más alto es con el método que utiliza todos los comentarios.

Para el método con No anécdotas el porcentaje más alto de respuestas Sí es de 72.22% que pertenece al tipo de búsqueda libre y de las personas con nivel de estudio no universitario. De hecho, podemos observar que para el método de todos los comentarios el rendimiento más alto es obtenido con el nivel de estudio universitario en los dos tipos de búsqueda.

El nivel de duda (No sé) aumenta para la búsqueda libre con todos los comentarios, lo que resulta entendible porque la búsqueda libre es más difícil de resolver para el sistema. Sin embargo, los porcentajes de respuestas de duda (No sé) bajan en la búsqueda libre para ambos niveles de estudio usando solo no anécdotas en comparación con usar todos los comentarios. Lo anterior vuelve a dar indicios de que eliminar las anécdotas parece eliminar fotografías menos relacionadas. Además, nuevamente se alcanza a ver que cada método beneficia a diferente perfil de personas, esto es, usar todos los comentarios tiene mejor rendimiento para universitarios y usar solo no anécdotas a los no universitarios, especialmente en búsqueda libre.

Método	Tipo de búsqueda	Nivel de estudio	Total de imágenes evaluadas	Sí	No	No sé
Todos	Asignada	Universitario	412 (100%)	309 (75%)	67 (16.26%)	36 (8.74%)
		No universitario	411 (100%)	258 (62.77%)	111 (27.01%)	42 (10.22%)
	Libre	Universitario	493 (100%)	318 (64.50%)	54 (10.95%)	121 (24.54%)
		No universitario	530 (100%)	267 (50.38%)	159 (30%)	104 (19.62%)
No anécdota	Asignada	Universitario	218 (100%)	152 (69.72%)	53 (24.31%)	13 (5.96%)
		No universitario	158 (100%)	108 (68.35%)	27 (17.09%)	23 (14.56%)
	Libre	Universitario	103 (100%)	60 (58.25%)	31 (30.10%)	12 (11.65%)
		No universitario	36 (100%)	26 (72.22%)	5 (13.89%)	5 (13.89%)

Tabla 4-9 Resultados por tipo de búsqueda y nivel de estudio

En el Anexo 2 se muestran las palabras buscadas y su evaluación para cada uno de los perfiles, con el objetivo de comparar resultados se presentan por pares de perfiles.

5. Conclusiones

5.1. Resumen de experimentos

En esta tesis se realizaron experimentos para dos tareas diferentes, la primera fue la recuperación automática de anécdotas y la segunda la recuperación automática de fotografías de la página de Facebook *La ciudad de México en el tiempo*.

El objetivo de los experimentos para la recuperación de anécdotas fue obtener un modelo clasificador de Anécdotas y No anécdotas de los comentarios de las fotografías. El método utilizado fue clasificación automática y la medida de evaluación fue *accuracy*, el modelo clasificador con la medida más alta (0.8021) tuvo las siguientes características:

- Vector TFIDF
- Utilizando unigramas
- Utilizando stop words
- Quitando acentos
- Truncando palabras
- Realizando una reducción de dimensionalidad a 300 componentes
- Utilizando el algoritmo de aprendizaje SVM con kernel rbf

Este modelo clasificó correctamente 322 comentarios como Anécdotas de un total 422 comentarios y 355 No anécdotas de 422.

Los siguientes experimentos fueron para la tarea de recuperación automática de fotografías a través de descriptores (palabras clave), donde el objetivo fue comparar el modo de generar los descriptores de las imágenes: con todos los comentarios o solo con los comentarios que no fueran anécdotas. El método fue basado en la recuperación de información y la forma de evaluar fue a través de porcentajes de respuestas Sí, No y No sé, según describiera una petición de búsqueda (un conjunto de palabras) a una fotografía. Estas evaluaciones las realizaron personas con diferentes perfiles de edad y nivel de estudio.

De forma general el método con mayor cantidad de respuestas Sí (donde las palabras clave sí describían la fotografía) fue donde los descriptores se generaban con los comentarios No anécdotas, donde las respuestas fueron:

- Sí con 67.18%
- No con 22.52%

- No sé con 10.29%

Para los descriptores generados con todos los comentarios el porcentaje de respuestas fue el siguiente:

- Sí con 62.41%
- No con 21.18%
- No sé con 16.41%

Otros patrones que pueden observarse de los resultados, aunque no de manera generalizada, son:

- Los porcentajes de duda (No sé) tienden a bajar en el método No anécdota
- Los porcentajes de respuesta negativa (No) suben en el método No anécdota
- El método No anécdota tiene mejor rendimiento en jóvenes y en no universitarios, mientras que el método Todos tiene mejor rendimiento en adultos mayores y en universitarios.

5.2.Revisión de preguntas, hipótesis y objetivos de investigación

Las preguntas de investigación que nos planteamos al inicio de esta tesis fueron:

- ¿Será posible la detección automática de recuerdos y anécdotas a partir de los comentarios generados en una página de Facebook?
- ¿Será posible la recuperación automática de imágenes (fotografías) publicadas en Facebook?

Según los resultados obtenidos, sí fue posible detectar automáticamente recuerdos y anécdotas a partir de los comentarios del sitio de Facebook *La Ciudad de México en el tiempo*; sin embargo, no fue posible hacerlo al 100% ya que solo se clasificaron correctamente 322 anécdotas de 422, y solo 355 no anécdotas de 422 (*accuracy* de 0.8021).

En cuanto a la recuperación de imágenes, sí fue posible hacerlo a través de palabras clave generadas de los comentarios. Según una evaluación manual del rendimiento, solo el 67.18% de las fotos estaban relacionadas con la petición del usuario cuando se usaron solo no anécdotas y 62.41% cuando se usaron todos los comentarios.

Las hipótesis planteadas el inicio de la tesis fueron:

- Sí es posible la detección automática de recuerdos y anécdotas a partir de los comentarios generados en una página de Facebook.
- Sí es posible la recuperación automática de imágenes (fotografías) publicadas en Facebook a través de la generación automática de descriptores.

En el caso de la detección automática de anécdotas, los resultados son muy alentadores (*accuracy* de 0.8021), por lo que podemos aceptar la hipótesis. Para el caso de la recuperación de imágenes, donde los resultados no fueron muy altos ni se comportaron igual para todos los perfiles de usuarios evaluadores, no es fácil aceptar la hipótesis; sin embargo, se trata de una tarea que implica procesamiento de lenguaje natural y resulta difícil para las computadoras por lo que tampoco podemos rechazarla. Lo anterior nos lleva a plantear una nueva hipótesis: sí es posible mejorar el rendimiento de la recuperación automática de imágenes, que podría ser comprobada en futuros trabajos.

Finalmente, lo objetivos principales de esta investigación fueron:

- Desarrollar un método para la detección automática de anécdotas en los comentarios de la página de Facebook *La ciudad de México en el tiempo*.
- Desarrollar un método para la recuperación automática de fotografías (imágenes) del sitio de Facebook *La ciudad de México en el tiempo* mediante la generación automática de descriptores.

Ambos objetivos se han cumplido obteniendo cada uno distintos niveles de rendimiento. Para el caso de la recuperación de imágenes consideraremos como mejor método el basado en No anécdotas solo por la evaluación general, ya que en el análisis de perfiles vimos que no era consistente.

El primer método para la detección automática de anécdotas en los comentarios de la página del Facebook *La ciudad de México en el tiempo* consistió en un proceso de clasificación automática tomando en cuenta diferentes características del texto (comentarios). Como se muestra en la Figura 5-1 el proceso tiene como entrada un corpus de comentarios, posteriormente se lleva a cabo el preprocesamiento para después crear un modelo clasificador con un algoritmo de aprendizaje. Finalmente se utiliza en modelo para clasificar en anécdotas y No anécdotas.

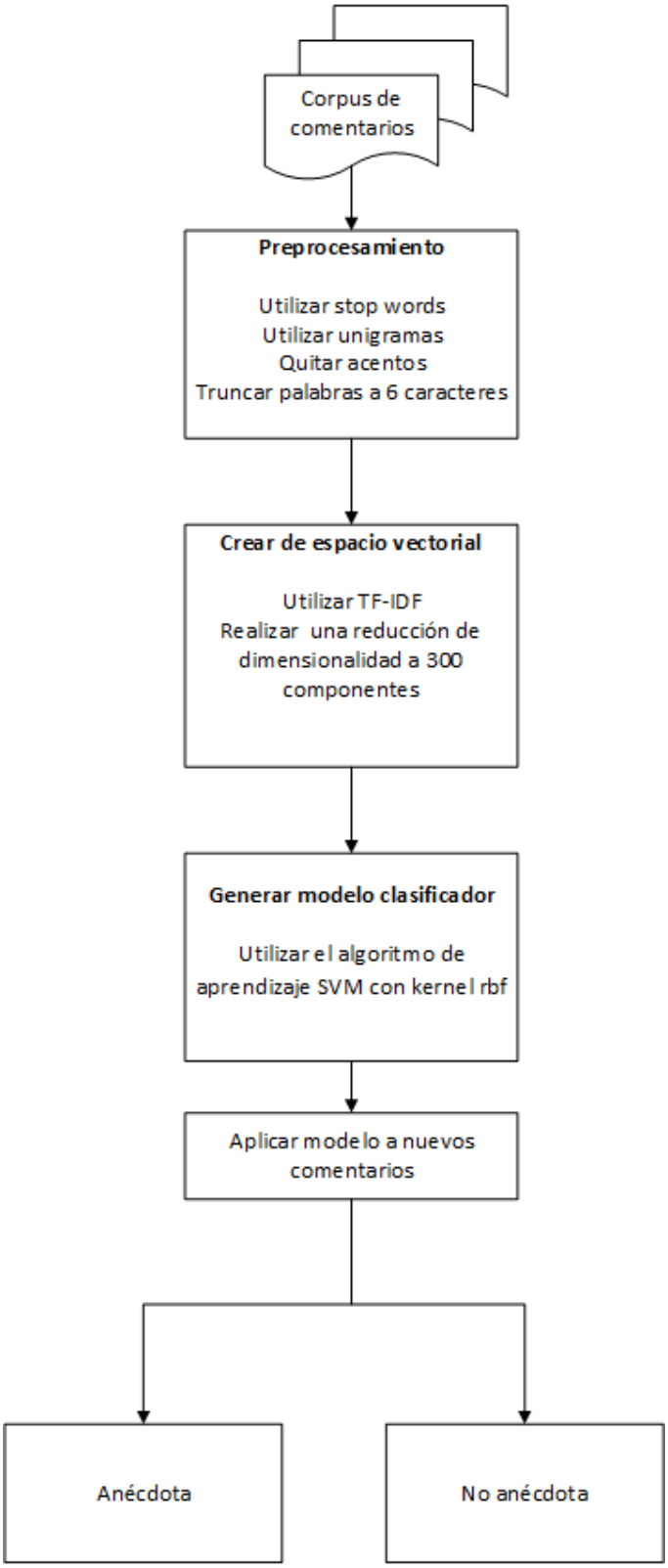


Figura 5-1 Método para la detección automática de anécdotas en los comentarios de la página del Facebook *La ciudad de México en el tiempo*

El segundo método para la recuperación automática de fotografías (imágenes) del sitio de Facebook *La ciudad de México en el tiempo* consistió en la generación automática de descriptores a partir de comentarios a las imágenes. Como se muestra en la Figura 5-2 se utilizan

los comentarios clasificados como No anécdotas, se obtiene la medida TF-IDF de cada palabra, utilizando estas medidas se obtiene el promedio y la desviación estándar. Si la medida TF-IDF de una palabra es mayor a la suma de estas cantidades es considerada descriptor de la imagen.

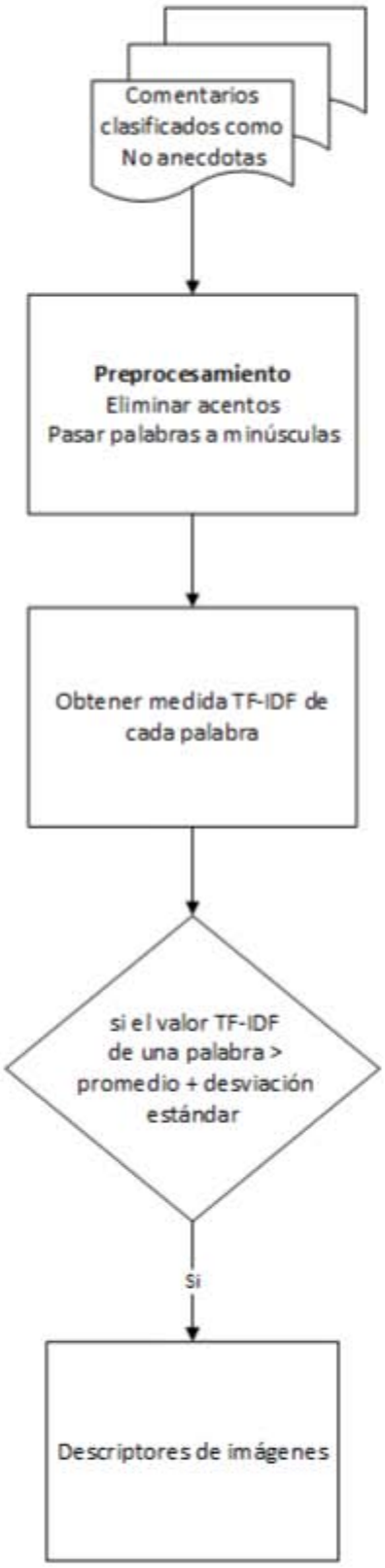


Figura 5-2 método para la recuperación automática de fotografías del sitio de Facebook *La ciudad de México en el tiempo*

5.3.Debilidades y fortalezas de los métodos propuestos

Una debilidad es que los métodos no recuperan al 100% las fotografías y anécdotas, por lo que podríamos estar perdiendo material interesante. Además, solo se está utilizando un corpus obtenido de un sitio de Facebook, por lo que los resultados podrían cambiar para otro corpus.

Una de las fortalezas es que se reduce el tiempo de búsqueda de imágenes específicas en el sitio *La Ciudad de México en el tiempo* por lo que los administradores del sitio podrían usar el método para cuando les pidan imágenes para libros.

5.4.Trabajo futuro

En la tarea de recuperación automática de anécdotas se puede mejorar el método para que recupere más anécdotas, para ello se podrían realizar otros experimentos con diferentes características, algoritmos de clasificación e incluso utilizar otras técnicas diferentes a la clasificación automática con el fin de mejorar el rendimiento del método propuesto.

En la tarea de recuperación automática de fotografías también se puede mejorar el método, si bien hasta ahora recupera fotografías de acuerdo a palabras clave, es posible hacer más amplio el método e incluir todos los álbumes y fotografías del sitio.

5.5.Comentarios finales

En general, realizar esta tesis fue un trabajo muy interesante y representó un reto ya que a pesar de que las redes sociales están presentes actualmente e interactuamos a través de ellas, no conocíamos algún método que se encargara específicamente de recuperar anécdotas y fotografías a partir de comentarios realizados por los mismos usuarios de la red. Nos enfrentamos a varios retos por tratar con redes sociales, uno de ellos fue la forma de escribir de los usuarios. Por ejemplo, uno de nuestros objetivos era recuperar anécdotas, para lo cual tomamos en cuenta diferentes características de una anécdota, pero al tratar con comentarios nos dimos cuenta que los algunos comentarios (no todos) tenían faltas de ortografía o cambiaban las palabras, además del uso de emoticones que tuvimos que tratar para poder obtener mejores resultados. Otro reto al que nos enfrentamos fue a la hora de evaluar las imágenes recuperadas, ya que fue difícil encontrar personas que estuvieran dispuestas a colaboraran en la evaluación, esto llevó a que fueron pocas personas involucradas en la evaluación.

Realizar esta tesis me dio la oportunidad de conocer diferentes formas de tratar los datos e información y específicamente la forma en la que el texto se convierte a un formato estructurado que puede ser procesado por sistemas de cómputo, además de diferentes técnicas de minería de texto que aplicamos para lograr los objetivos planteados.

Es muy interesante como utilizamos técnicas de minería de texto para tratar los comentarios de la red social, ya que actualmente se procesan grandes cantidades de información y la minería de textos juega un papel muy importante no solo en campos científicos, si no que en las empresas se utilizan técnicas de minería de texto o minería de datos para mejorar ventas o encontrar patrones que mejoren sus procesos. En este caso tratamos con datos no estructurados que podemos encontrar muy frecuentemente en texto, correo electrónico, páginas web, etc. Finalmente, podemos decir que el tratamiento de esta información puede mejorar los procesos en diferentes organizaciones por lo que las áreas de minería de textos y procesamiento de lenguaje natural juegan un papel importante en la informática.

6. Anexos

Anexo 1

ID	Tipo	N-gramas	Algoritmo	Quitar SW	Quitar acentos	Truncar	Red Dim	Etiquetas POS	Kernel	normalizacion	C	Accuracy
1	Conteo	1	SVM	No	Sí	Sí	300	NA	linear	log2	10	0.658767773
2	Conteo	1	SVM	No	Sí	Sí	300	NA	rbf	log2	1000	0.664691943
3	Conteo	1	SVM	No	Sí	Sí	300	NA	linear	None	1	0.771327014
4	Conteo	1	SVM	No	Sí	Sí	300	NA	rbf	None	10	0.797393365
5	Conteo	1	SVM	No	Sí	Sí	200	NA	linear	log2	1	0.656398104
6	Conteo	1	SVM	No	Sí	Sí	200	NA	rbf	log2	1000	0.662322275
7	Conteo	1	SVM	No	Sí	Sí	200	NA	linear	None	1	0.785545024
8	Conteo	1	SVM	No	Sí	Sí	200	NA	rbf	None	10	0.800947867
9	TFIDF	1	SVM	No	Sí	Sí	200	NA	rbf	None	10	0.800947867
10	TFIDF	1	SVM	No	Sí	Sí	300	NA	rbf	None	10	0.802132701
11	Binario	1	SVM	No	Sí	Sí	200	NA	rbf	None	10	0.797393365
12	Binario	1	SVM	No	Sí	Sí	300	NA	rbf	None	10	0.793838863
13	Conteo	1	SVM	No	Sí	Sí	No	NA	rbf	None	100	0.791469194
14	TFIDF	1	SVM	No	Sí	Sí	No	NA	rbf	None	10	0.778436019
15	Conteo	1	SVM	NA	NA	NA	No	Verbos todos	rbf	None	1000	0.732227488
16	Conteo	1	SVM	NA	NA	NA	No	Verbos todos	linear	None	1	0.735781991
17	Conteo	1	SVM	NA	NA	NA	No	Verbos pasado	rbf	None	1000	0.723933649
18	Conteo	1	SVM	NA	NA	NA	No	Verbos pasado	linear	None	1000	0.723933649
19	Binario	1	SVM	NA	NA	NA	No	Verbos todos	rbf	None	100	0.739336493
20	Binario	1	SVM	NA	NA	NA	No	Verbos pasado	rbf	None	1000	0.719194313
21	TFIDF	1	SVM	NA	NA	NA	No	Verbos todos	rbf	None	1	0.728672986
22	TFIDF	1	SVM	NA	NA	NA	No	Verbos pasado	rbf	None	100	0.726303318
23	Binario	1	SVM	NA	NA	NA	No	Todas	rbf	None	10	0.783175355

ID	Tipo	N-gramas	Algoritmo	Quitar SW	Quitar acentos	Truncar	Red Dim	Etiquetas POS	Kernel	normalizacion	C	Accuracy
24	TFIDF	1	SVM	NA	NA	NA	No	Todas	rbf	None	10	0.760663507
25	COUNT	1	SVM	NA	NA	NA	No	Todas	rbf	None	100	0.764218009
39	Conteo	1	SVM	No	Sí	No	200	NA	rbf	None	10	0.787914692
40	TFIDF	1	SVM	No	Sí	No	300	NA	rbf	None	10	0.79028436
41	Binario	1	SVM	No	Sí	No	200	NA	rbf	None	1000	0.783175355
42	Conteo	1	SVM	No	No	Sí	200	NA	rbf	None	10	0.786729858
43	TFIDF	1	SVM	No	No	Sí	300	NA	rbf	None	10	0.802132701
44	Binario	1	SVM	No	No	SI	200	NA	rbf	None	10	0.786729858
45	Conteo	1	SVM	No	No	No	200	NA	rbf	None	10	0.780805687
46	TFIDF	1	SVM	No	No	No	300	NA	rbf	None	10	0.787914692
47	Binario	1	SVM	No	No	No	200	NA	rbf	None	100	0.777251185
48	Conteo	2	SVM	No	Sí	Sí	200	NA	rbf	None	1000	0.708530806
49	TFIDF	2	SVM	No	Sí	Sí	300	NA	rbf	None	10	0.708530806
50	Binario	2	SVM	No	Sí	Sí	200	NA	rbf	None	1000	0.704976303
51	Conteo	3	SVM	No	Sí	Sí	200	NA	rbf	None	1000	0.581753555
52	TFIDF	3	SVM	No	Sí	Sí	300	NA	rbf	None	100	0.612559242
53	Binario	3	SVM	No	Sí	Sí	200	NA	rbf	None	1000	0.582938389
54	Conteo	1,2,3	SVM	No	Sí	Sí	200	NA	rbf	None	10	0.79028436
55	TFIDF	1,2,3	SVM	No	Sí	Sí	300	NA	rbf	None	10	0.781990521
56	Binario	1,2,3	SVM	No	Sí	Sí	200	NA	rbf	None	10	0.793838863
57	Conteo	1	SVM	SI	Sí	SI	200	NA	rbf	None	100	0.748815166
58	TFIDF	1	SVM	SI	Sí	SI	300	NA	rbf	None	100	0.755924171
59	Binario	1	SVM	SI	Sí	SI	200	NA	rbf	None	1000	0.731042654
60	Conteo	1	MultinomialNB	No	Sí	Sí	No	NA	NA	None	NA	0.776066351

ID	Tipo	N-gramas	Algoritmo	Quitar SW	Quitar acentos	Truncar	Red Dim	Etiquetas POS	Kernel	normalizacion	C	Accuracy
61	TFIDF	1	MultinomialNB	No	Sí	Sí	No	NA	NA	None	NA	0.747630332
62	Binario	1	MultinomialNB	No	Sí	Sí	No	NA	NA	None	NA	0.780805687
63	Conteo	1	MultinomialNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.719194313
64	Conteo	1	MultinomialNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.684834123
65	Binario	1	MultinomialNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.719194313
66	Binario	1	MultinomialNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.665876777
67	TFIDF	1	MultinomialNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.712085308
68	TFIDF	1	MultinomialNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.686018957
69	Binario	1	MultinomialNB	NA	NA	NA	No	Todas	NA	None	NA	0.786729858
70	TFIDF	1	MultinomialNB	NA	NA	NA	No	Todas	NA	None	NA	0.748815166
71	COUNT	1	MultinomialNB	NA	NA	NA	No	Todas	NA	None	NA	0.757109005
72	Conteo	1	GaussianNB	No	Sí	Sí	No	NA	NA	None	NA	0.657582938
73	TFIDF	1	GaussianNB	No	Sí	Sí	No	NA	NA	None	NA	0.654028436
74	Binario	1	GaussianNB	No	Sí	Sí	No	NA	NA	None	NA	0.659952607
75	Conteo	1	GaussianNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.527251185
76	Conteo	1	GaussianNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.662322275
77	Binario	1	GaussianNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.528436019
78	Binario	1	GaussianNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.650473934
79	TFIDF	1	GaussianNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.524881517
80	TFIDF	1	GaussianNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.662322275
81	Binario	1	GaussianNB	NA	NA	NA	No	Todas	NA	None	NA	0.672985782
82	TFIDF	1	GaussianNB	NA	NA	NA	No	Todas	NA	None	NA	0.631516588
83	COUNT	1	GaussianNB	NA	NA	NA	No	Todas	NA	None	NA	0.626777251
84	Conteo	1	BernoulliNB	No	Sí	Sí	No	NA	NA	None	NA	0.760663507

ID	Tipo	N-gramas	Algoritmo	Quitar SW	Quitar acentos	Truncar	Red Dim	Etiquetas POS	Kernel	normalizacion	C	Accuracy
85	TFIDF	1	BernoulliNB	No	Sí	Sí	No	NA	NA	None	NA	0.760663507
86	Binario	1	BernoulliNB	No	Sí	Sí	No	NA	NA	None	NA	0.760663507
87	Conteo	1	BernoulliNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.735781991
88	Conteo	1	BernoulliNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.722748815
89	Binario	1	BernoulliNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.735781991
91	TFIDF	1	BernoulliNB	NA	NA	NA	No	Verbos todos	NA	None	NA	0.735781991
92	TFIDF	1	BernoulliNB	NA	NA	NA	No	Verbos pasado	NA	None	NA	0.722748815
93	Binario	1	BernoulliNB	NA	NA	NA	No	Todas	NA	None	NA	0.747630332
94	TFIDF	1	BernoulliNB	NA	NA	NA	No	Todas	NA	None	NA	0.747630332
95	COUNT	1	BernoulliNB	NA	NA	NA	No	Todas	NA	None	NA	0.747630332

Anexo 2

Joven Universitario

Perfil 01

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	10	1	1
asignada	estadio azteca	29	17	3	9
asignada	monumento revolucion	18	11	2	5
asignada	torre latino	37	34	0	3
asignada	zocalo catedral	15	14	1	0
asignada	palacio bellas artes	27	19	6	2
Libre	reforma	121	85	2	34
Libre	tacubaya	62	28	4	30
Libre	indios verdes	20	12	4	4
Libre	metro Hidalgo	4	1	0	3
Libre	xochimilco	72	35	10	26
Libre	insurgentes glorieta	22	12	3	7
Libre	villa basilica	13	11	1	1

Perfil 07

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	8	7	0	1
asignada	estadio azteca	15	10	0	5
asignada	monumento revolucion	9	9	0	0
asignada	torre latino	25	22	2	1
asignada	zocalo catedral	7	7	0	0
asignada	palacio bellas artes	14	12	1	1
libre	castillo chapultepec	5	4	0	1
libre	diana cazadora	6	1	0	5
libre	templo mayor	7	5	1	1
libre	six flags	1	1	0	0
libre	plaza garibaldi	2	1	0	1
libre	san ildefonso	1	1	0	0

Joven No Universitario

Perfil 02

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	8	2	1
asignada	estadio azteca	29	18	11	0
asignada	monumento revolucion	18	9	9	0
asignada	torre latino	37	18	18	0
asignada	zocalo catedral	15	14	1	0
asignada	palacio bellas artes	27	11	15	0
libre	moneda	22	8	14	0
libre	angel independencia	4	4	0	0
libre	templo mayor	13	4	9	0
libre	castillo chapultepec	10	3	7	0
libre	diana cazadora	9	2	7	0
libre	tlatelolco	66	29	36	0
libre	banco mexico	10	5	1	3
libre	politecnico	16	11	0	5
libre	politecnico	16	11	0	5
libre	calle madero agua iglesias arte museos	0	0	0	0
libre	agua	154	25	9	3
libre	iglesias basilica	1	1	0	0

Perfil 08

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	estadio azteca	15	0	0	0
asignada	estadio azteca	15	0	0	0
asignada	estadio azteca	15	9	2	2
libre	calle corregidora	3	2	0	1
asignada	monumento revolucion	9	8	0	1
asignada	torre latino	25	14	6	5
asignada	zocalo catedral	7	7	0	0
asignada	palacio bellas artes	14	9	4	1
asignada	palacio deporte	8	5	2	1
libre	museo el chopo	1	1	0	0
libre	teatro insurgente	3	1	0	2
libre	bosques arboles	0	0	0	0
libre	niños edificios	2	1	1	0
libre	aviones	20	16	2	1

Adulto Universitario

Perfil 03

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	10	2	0
asignada	estadio azteca	29	20	7	1
asignada	monumento revolucion	18	12	2	4
asignada	torre latino	37	24	9	4
asignada	zocalo catedral	15	13	1	1
asignada	palacio bellas artes	27	17	9	1
libre	canales xochimilco	19	12	2	5
libre	calles tacubaya	1	1	0	0
libre	basilica	30	22	8	0
libre	museo antropologia	8	6	2	0
libre	polanco	38	22	8	8
libre	eje central	34	23	9	2

Perfil 09

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	8	6	0	2
asignada	monumento revolucion	9	9	0	0
asignada	torre latino	25	16	7	2
asignada	zocalo catedral	7	7	0	0
asignada	palacio bellas artes	14	8	5	1
libre	pulqueria cantina	1	1	0	0
libre	merced barrio	3	1	0	2
libre	metro transporte	11	4	7	0
libre	cantina	19	6	11	2
libre	circo	9	4	5	0
libre	tapo	2	1	1	0
libre	aeropuerto	31	26	5	0

Adulto No universitario

Perfil 04

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	11	0	1
asignada	estadio azteca	29	17	10	2
asignada	monumento revolucion	18	10	7	1
asignada	torre latino	37	23	7	7
asignada	zocalo catedral	15	13	1	1
asignada	palacio bellas artes	27	18	8	1
libre	cerro tlahuac	3	2	0	1
libre	xochimilco	72	38	15	19
libre	tulyehualco	6	0	4	2
libre	tacubaya	62	22	17	23
libre	museo antropologia	8	4	2	2
libre	castillo chapultepec	10	5	4	1

Perfil 10 Sin evaluación

Adulto mayor universitario

Perfil 05

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	11	0	1
asignada	estadio azteca	29	20	7	2
asignada	monumento revolucion	18	11	6	1
asignada	torre latino	37	32	4	0
asignada	zocalo catedral	15	13	2	0
asignada	palacio bellas artes	27	21	5	1
libre	dulces chocolates	1	1	0	0
libre	parques jardines	1	1	0	0
libre	tranvia metro	23	21	1	0
libre	lagos inundaciones	3	2	0	1
libre	derrumbes	2	2	0	0
libre	mercados tepito	1	1	0	0
libre	camion transporte	16	16	0	0
libre	loteria nacional caballito	2	2	0	0

Perfil 11

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	8	2	6	0
asignada	estadio azteca	15	6	9	0
asignada	monumento revolucion	9	4	5	0
asignada	torre latino	25	12	13	0
asignada	zocalo catedral	7	4	2	0
asignada	palacio bellas artes	14	11	3	0
libre	revolucion museo	1	1	0	0
libre	fotografia construccion	2	2	0	0
libre	personas historia	1	1	0	0
libre	costumbres tiempo	1	0	1	0

Adulto mayor no universitario

Perfil 06

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	12	9	1	2
asignada	estadio azteca	29	17	2	10
asignada	monumento revolucion	18	7	7	4
asignada	torre latino	37	23	6	8
asignada	zocalo catedral	15	13	1	1
asignada	palacio bellas artes	27	19	5	3
libre	xochimilco	72	40	10	22
libre	Basilica	30	20	6	4
libre	chapultepec	99	55	27	17
libre	plaza garibaldi	3	3	0	0

Perfil 12

Tipo de busqueda	Palabras	Fotos presentadas	Si	No	No sé
asignada	palacio deporte	8	6	0	2
asignada	estadio azteca	15	10	3	2
asignada	monumento revolucion	9	7	0	2
asignada	torre latino	25	12	7	4
asignada	zocalo catedral	7	6	0	1
asignada	zocalo catedral	7	6	0	1
asignada	palacio bellas artes	14	9	3	1
libre	edificio correos	2	1	0	1
libre	palacio nacional	2	1	1	0
libre	plaza toros	4	3	1	0

7. Referencias

AMIPCI. (2015). *11° estudio sobre los hábitos de los usuarios de internety en México 2015.* Obtenido de https://www.amipci.org.mx/images/AMIPCI_HABITOS_DEL_INTERNAUTA_MEXICANO_2015.pdf

Betancourt, G. A. (2005). Las máquinas de soporte vectorial (SVMs). *Scientia et Technica*, 1(27).

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs, 41*(6), 391--407.

Elmasri, R., & B. Navathe, S. (2007). *Fundamentos de sistemas de bases de datos, 5ta edición.* Madrid: Pearson Educación S.A.

Gelbukh, A., & Sidorov, G. (2010). *Procesamiento automático del español con enfoque en recursos léxicos grandes.* México, DF: Instituto Politécnico Nacional.

Gorunescu, F. (2011). *Data Mining: Concepts, models and techniques* (Vol. 12). Berlin Heidelberg: Springer Science & Business Media.

H. Witten, I., Frank, E., & A. Hall, M. (2011). *Data Mining, Practical Machine Learning Tools and Techniques, Third Edition.* Elsevier.

Harrington, P. (2012). *Machine learning in action.* Manning Publications.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval.* Cambridge University Press.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval.* Cambridge University Press.

Moreno, A. (1994). *Aprendizaje automático.* Barcelona: Edicions UPC.

Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825-2830.

Pérez Gay, de Mauleón, & Villasana Suaveza. (2013). *Ciudad Sueño y Memoria.*

Suárez, C. (2014). Tutorial sobre Máquinas de Vectores Soporte (SVM).

Tan, A. H. (1999). Text mining: The state of the art and the challenges. . *In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, (Vol. 8, pp. 65-70).

Weiss S.M, Indurkhya, N, Zhang, T., & Demerau, F. (2010). *Text mining: predictive methods for analyzing unstructured information.* Springer Science & Business Media.