INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Recomendación de sitios turísticos con base en la detección de sentimientos de emociones recuperadas en Twitter

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Ing. Vanessa Borráz López

DIRECTOR DE TESIS:

Dra. Sandra Dinora Orantes Jiménez

CDMX., JUNIO 2019

SIP-14

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 12:00 horas del día 06 del mes de junio de 2019 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Recomendación de sitios turísticos con base en la detección de sentimientos de emociones recuperadas en Twitter"

Presentada por el alumno:

BORRÁZ	**LÓPEZ**	**VANESSA**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

B	1	6	0	6	2	8

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de Tesis

Dra. Sandra Dinora Orantes Jiménez

Dr. Grigori Sidorov

Dr. Moisés Salinas Rosales

Dra. GuoHua Sun

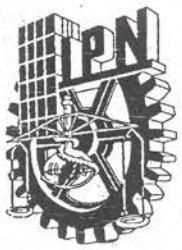
Dra. Graciela Vázquez Álvarez

Dr. Rolando Menchaca Méndez

PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Dr. Marco Antonio Ramírez Salinas



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la **Ciudad de México** el día **10** del mes **junio** del año **2019**, la que suscribe **Vanessa Borráz López,** alumna del programa de **Maestría en Ciencias de la Computación** con número de registro **B160628**, adscrito al **Centro de Investigación en Computación**, manifiesta que es autora intelectual del presente trabajo de Tesis bajo la dirección de la **Dra. Sandra Dinora Orantes Jiménez** y cede los derechos del trabajo intitulado **Recomendación de sitios turísticos con base en la detección de sentimientos de emociones recuperados de Twitter**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección vanessa.borrazl@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Vanessa Borráz López

Resumen

Esta investigación se realizó con el apoyo de herramientas y algoritmos como: API Stream de Twitter, base de datos NoSQL MongoDB, algoritmo *Support Vector Machine* SVM (tf*idf, bag of words), K-Fold Cross Validation, el micro *framework* Flask y la base de datos DBpedia, trabajando en conjunto sobre un sistema operativo Linux.

Estas herramientas en conjunto realizaron un estudio del comportamiento de la información de usuarios viajeros en Twitter, para generar un reporte de ayuda en la toma de decisiones en futuros viajes del turismo mexicano de una forma más fácil y organizada, de la siguiente manera:

API Stream operó en la recopilación de *tweets* en tiempo real de cuentas de viajeros de Twitter, para la generación de corpus durante 6 meses; para ello, se necesitó de una base de datos MongoDB NoSQL, para el manejo y guardado de datos JSON.

SVM se considera una parte fundamental para esta tesis, debido al uso de una gran cantidad de *tweets*; SVM ayudó a verificar el trabajo humado (Aprendizaje Supervisado) y así, con mayor certeza clasificar *tweets* como se deseaba; lográndose hacer más legible el contenido de los *tweets*, debido a la normalización de datos que se aplicó para el mejor funcionamiento del algoritmo.

Para darle un uso práctico y claro a éste trabajo, se decidió mostrar la información procesada en forma de reporte Web; lo cual llevó al uso de otras herramientas como la base de datos BDPedia (se usó para vincular la información de *tweets* con estados mexicanos u otros países o lugares) y el micro *framework* Flask (se usó para consultar y mostrar la información de una forma ágil a los usuarios finales).

Finalmente, se genera “Reporte para Viajeros MX” que permite tener una idea real y resumida de las vivencias que han tenido personas en el mundo, que comparten sus experiencias en las redes sociales.

Los resultados finales muestran que a partir de un corpus generado de Twitter se obtuvo un reporte web, en el cual se observa que empleando el algoritmo SVM con aprendizaje supervisado se obtiene la información de mayor utilidad para usuarios que tomen como guía éste trabajo para conocimiento propio o bien para planificar sus viajes en México. La clasificación y entrenamiento con SVM y el empleo de la base de datos DBpedia para organizar la información procesada por estados mexicanos, muestran finalmente un reporte con recomendaciones organizadas y con consultas rápidas de recomendaciones a destinos mexicanos de su elección.

Como trabajos futuros, empleando la misma metodología, se podría mejorar éste trabajo no limitando la información en México, si no tal vez en los destinos turísticos mayormente recomendados a nivel mundial e incluso obteniendo el corpus de una manera distinta a la que se obtuvo en éste trabajo o tomando en cuenta el procesamiento de imágenes o video de modo que se obtenga mayor información de utilidad para tener un panorama más amplio de las opiniones de ayuda de viajeros de Twitter.

Abstract

This investigation was realized with tools and algorithms like Stream API, MongoDB database NoSQL, SVM algorithm (tf*idf, bag of words), K-Fold Cross Validation, the micro *framework* Flask and DBpedia database, all of this working together in an operative system Linux.

The above tools were used in an information analysis of traveler users that was extracted from Twitter to generate an organized, easy and helping report in the decisions making in future trips of Mexican tourism like follow:

Stream API worked in real time for collect and save each event executed of the 3 different Twitter accounts for a period of 6 months in MongoDB.

SVM is considerate a fundamental part for this thesis work because it helps in the machine learning process using the obtained corpus; SVM with K-fold cross validation help to verify a good supervised learning work and thus, whit greater certainty categorize or section the tweets as desired; getting more legible the *tweets* content for the use of normalization in the corpus.

Getting a practice use and clear for this work, the information processed was showed like a report; this help to use other tools as BDpedia database (It was use to link tweets with Mexican states and other countries or places) and the micro framework Flask (This was use to consult and show information more fast and agile for the end users).

Finally, is generated a report call “Reporte para Viajeros MX” that It has a resume and real idea of people experiences who have had traveling around the world and they decided share it with all the world.

The final results show that from a generated Twitter corpus, a web report was obtained; in which it´s observed that using an SVM algorithm with supervised learning we can obtain the usefullest information for users that use this information for their travels or own knowledge. The classification, the training with SVM and the use of DBpedia database for organize processed Mexican information, finally show a report with organize recommendations and fast consultations of Mexican destinations of your choice.

Like a future works, thinking in the same idea and with the same technology this work could be better without limiting the information to Mexico, probably could be addressed to the places with more tourism in the world, inclusive getting the corpus of a distinct manner taking into account the information of images and videos for generation of the corpus.

Agradecimientos

Al Instituto Politécnico Nacional por haberme otorgado el privilegio de realizar mis estudios de posgrado y de pertenecer a una de las instituciones pilares de la educación en México.

Al Centro de Investigación en Computación por haberme dado la oportunidad de cursar este posgrado y por el apoyo académico y tecnológico para la elaboración de este trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) que, como organización, me brindó el apoyo económico necesario a través de la beca de posgrado para la realización de este trabajo.

A mi directora de tesis Sandra Dinora Orantes Jiménez por su apoyo, tiempo y dedicación en la revisión del presente trabajo. Y sobre todo por su empatía y confianza hacia mi persona.

A mis sinodales por sus enseñanzas académicas que ayudaron en la realización de éste trabajo, por su tiempo invertido y dedicación en la revisión y entendimiento de éste trabajo. Gracias por sus observaciones y consejos para una mejora continua en mi desempeño profesional.

Dedicatoria

Debo mencionar y agradecer a mi familia que nuevamente han sido parte fundamental en la obtención de éste grado, especialmente a mi madre. Pero no sería honesta si menciono más de una dedicatoria en ésta tesis, desde el comienzo de éste trabajo no deje de tener en mi mente y corazón a una personita que me ayudó en el impulso de querer obtener un mayor grado en mi vida profesional; motivo por el cual, es mi deseo y oportunidad de recompensar su esfuerzo y valentía.

A MI HIJO

Gracias por ser esa personita madura que comprende que mamá quiere seguir adelante y desea de corazón dejar ese ejemplo en lo que más ama. Por nosotros comencé y terminé éste posgrado, deseando que no haya sido en vano el esfuerzo que decidí realizar. Eres mi mayor inspiración, mi fuerza, mi deseo de que logres muchos más objetivos que yo. No será éste el mejor trabajo de tesis, pero sin duda éste título de Maestría en Ciencias de la Computación vendrá siempre a mi mente junto a tu esfuerzo y tu gran corazón.

Con amor, para ti Caleb.

"*Never Give Up, Beautiful Boy*"

Contenido

Lista de figuras

Lista de tablas

1) CAPITULO I. Introducción

Las Redes Sociales son la evolución de las maneras tradicionales de comunicación del ser humano, que han avanzado con el uso de nuevos canales y herramientas y que se basan en la co-creación, conocimiento colectivo y confianza generalizada. Dentro de estos nuevos canales pueden encontrarse multitud de clasificaciones diferentes como: *blogs,* agregadores de noticias, *wikis*, etc., que usados conjuntamente permiten una potencial interacción con miles de personas con inquietudes comunes.

Por consiguiente, los medios sociales se convierten en una buena herramienta ya que, a la hora de buscar información en Redes Sociales debe hacerlo de una manera eficaz y para ello, es imprescindible tener una técnica claramente definida que ayude a seguir la línea marcada y así, no dar vueltas de más, que hagan perder tiempo y dinero.

Por otra parte, el análisis de sentimientos es una tarea incluida en el ámbito del Procesamiento de Lenguaje Natural o NLP (Natural Language Processing), del Análisis de Textos y de la Lingüística Computacional y su objetivo es encontrar contenido subjetivo en los textos de entrada; por otra parte, busca extraer opiniones y la polaridad de estas de uno o más documentos, mediante la extracción de una serie de características que determinen cuán positivo o negativo es el texto.

Algunas tareas para las que puede emplearse este campo son, por ejemplo, encontrar la valoración (puntuación) de un restaurante o película con base en la polaridad extraída de diferentes críticas y reseñas que existan al respecto; es así, como puede emplearse particularmente para encontrar opiniones buenas y malas en una red social como Twitter, sobre distintos temas como, por ejemplo, destinos turísticos a los que los usuarios han viajado y de los cuáles, comparten sus experiencias.

Así también, el análisis de polaridad sobre Twitter pretende establecer una opinión generalizada de una entidad concreta, ya que por su versatilidad y variedad de usuarios que cada día publican contenido; convierten a la red social en el lugar perfecto para extraer la información que permita establecer la opinión mayoritaria acerca de prácticamente cualquier tema. Por otra parte, el análisis de polaridad permite estudiar cómo afectan diversos factores sociales a la opinión que la sociedad tiene acerca de un tema; por ejemplo, cómo modifica una determinada campaña política a la valoración de las personas respecto a un partido concreto. Al estudio de esta evolución se le denomina Análisis de Tendencias.

Por último, los sistemas de resumen automático pueden resultar una herramienta muy útil a la hora de complementar al análisis de sentimientos, ya que son herramientas que buscan extraer el contenido más relevante de un conjunto de documentos. El resumen generado automáticamente, debe

representar la información contenida en el texto sin incluir redundancias o partes del texto poco importantes. Estos sistemas se pueden emplear para extraer el contenido más representativo de los documentos de los que se esté realizando el análisis de polaridad. De este modo, cuando se trata de Twitter, no solo se obtiene la polaridad, sino también el conjunto de *tweet* que representan las opiniones.

Por consiguiente, para esta investigación se considera la implementación de algoritmos de análisis de sentimientos en *tweets*, donde los comentarios de los usuarios viajeros de Twitter son la fuente principal de alimentación para el procesamiento de información (corpus); lo cual, pretende tomarse como base para el proceso de generación del "Reporte para Viajeros MX", en el cual se obtendrán recomendaciones de utilidad, empleando como caso de estudio viajar a algún destino mexicano, ya sea con o sin planes de turismo; obteniendo información de que lugares, restaurantes, cosas, etc. son recomendadas y cuáles no.

1.1 Antecedentes

El auge de las Redes Sociales a través de Internet (RSI) en los últimos años, como *Facebook, Twitter, Google+, YouTube, LinkedIn o Pinterest*, han cambiado la forma en que las personas se comunican a través de Internet. Las empresas, conscientes de que sus clientes son parte activa de las RSI, han incrementado el interés de los encargados del área de *comercialización* para explorarlas como una nueva herramienta de *marketing*. Dada la novedad del fenómeno y su popularidad, muchas personas y empresas han comenzado a utilizar las RSI como una herramienta donde se puede extraer información para el análisis de tendencias, algunas incluso sin ningún tipo de estrategia.

Twitter es una herramienta social de *microblogging* en la que los usuarios pueden publicar contenido textual de hasta 140 caracteres; desde su nacimiento en 2006, Twitter se ha convertido en una de las plataformas más empleadas para compartir contenido de actualidad, como noticias o eventos en tiempo real, que además permite opinar de manera breve y concisa sobre prácticamente cualquier tema. En marzo del 2018, Twitter cuenta con 328 millones de usuarios activos los cuales, producen grandes cantidades de información de manera incesante. [23]

Existen aplicaciones enfocadas al turismo basadas en blogs con 7 destinos populares en EU y Europa, donde comparan 3 algoritmos de aprendizaje de máquina supervisado, los cuales son: Naive Bayes, SVM y el modelo n-gram para la clasificación de sentimientos.

1.2 Planteamiento del problema

La idea de ésta investigación surge, centrándose y tomando como caso de estudio la recomendación de sitios turísticos mexicanos, debido que, en la actualidad no se tiene un proceso que permita recuperar información de viajes en tiempo real, que sirva como base para un acercamiento actualizado a lugares turísticos y a opiniones, que puedan llevar a tomar decisiones respecto a la planeación adecuada de una vacación, visita de negocios a un determinado lugar o simplemente para obtener información sobre México.

1.3 Objetivos

1.3.1 Objetivo General

El objetivo general es generar un reporte de recomendaciones de sitios turísticos de México obteniendo un corpus a partir de publicaciones y comentarios de 3 cuentas de viajeros de Twitter en tiempo real, a través de un análisis y clasificación de sentimientos empleando el algoritmo SVM con aprendizaje supervisado, para obtener información de México de una forma más rápida, fácil y organizada que ayude la toma de decisiones de futuros viajes o bien para conocimiento propio.

1.3.2 Objetivos Específicos

- Establecer un estado del arte adecuado de trabajos relacionados al análisis de sentimientos que permitan sustentar y enriquecer la presente investigación y los resultados obtenidos.
- Seleccionar cuentas en Twitter enfocadas al sector turismo para realizar un análisis de recuperación de información.
- Establecer un Corpus de por lo menos 3000 tweets.
- Diseñar un proceso de clasificación de sentimientos que proporcione una base para informar o ayudar en el análisis y toma de decisiones en el sector turismo en México.
- Comprobar si Twitter refleja opiniones adecuadas que sirvan como base para informar o ayudar en el análisis y toma de decisiones de viajeros.
- Mostrar de una forma estructurada un reporte web para viajeros.

1.4 Justificación

Con la participación de usuarios en redes sociales, éstas se han convertido en el foco de muchos empresarios para mejorar y/o vender sus productos y realizar estudios de mercado; lo cual, permite encontrar un área, que permita realizar un proceso que ayude a tener una mejor experiencia y a beneficiarse con una base real para la toma de decisiones respecto a viajes que desee o necesite realizar.

Un estudio a inicio del año 2018 realizado por “We are Social” y “Hootsuite” menciona que en abril de 2018 la población mundial era de 7.6 mil millones de personas. Los usuarios de redes sociales crecieron en 121 millones entre el segundo trimestre de 2017 y el tercer trimestre del mismo año [22].

Se realizó un estudio que muestra la cantidad de usuarios activos en un mes por las diferentes redes sociales que se muestra en la gráfica de la Figura 1.1.

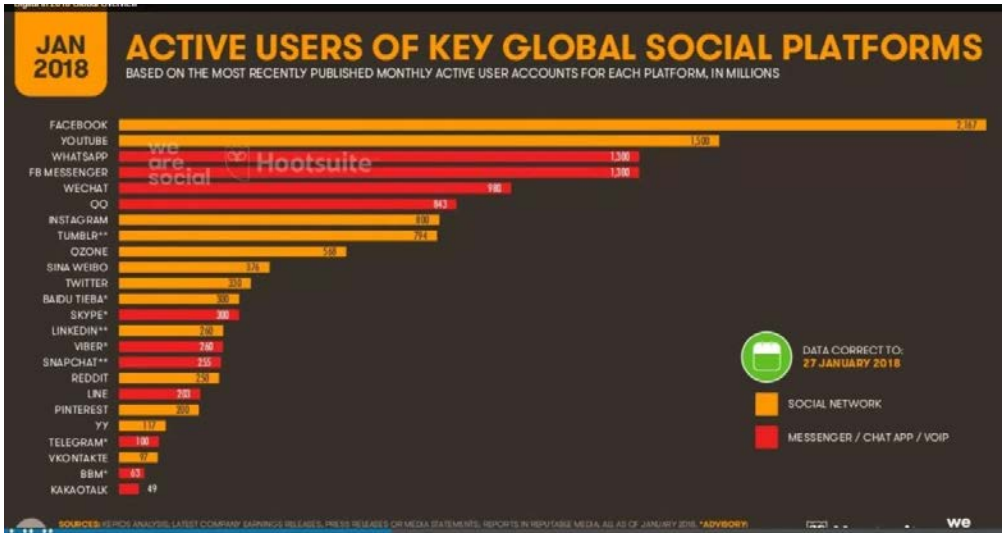


Fig. 1-1 Cantidad de usuarios activos por mes en diferentes redes sociales [22]

En este estudio se observa que Twitter cuenta con más 320 millones de usuarios activos en un mes. Esta red social es una de las que tiene un crecimiento más lento, sin embargo, se vuelve indispensable para la estrategia de *social media marketing* debido a varios aspectos:

- Su fortaleza en la información en tiempo real.
- Se puede convertir en el medio de comunicación oficial de las marcas.
- Es una red social que es mayoritariamente pública lo que permite que a las marcas realizar escucha social a través de ella.

Por lo tanto, se hará uso de Twitter para obtener y analizar *tweets* de distintos usuarios enfocados al turismo en México para generar un corpus con opiniones de éstos usuarios y posteriormente aplicar un procesamiento de éste corpus con algoritmos de lenguaje natural. SVM es una buena herramienta para formular un proceso que sirva de apoyo para obtener una mejor experiencia en viajes partiendo de un corpus obtenido de Twitter, que para esta investigación es de suma utilidad ya que se centra en mostrar por medio de una página web recomendaciones buenas o malas de lugares en México según las experiencias de usuario viajero de Twitter.

1.5 Beneficios esperados

Al día se realizan 500 millones de *tweets* de 100 millones de usuarios activos en Twitter. Los países con más usuarios en Twitter, son EU (67 millones de usuarios), Brasil (27.7 millones de usuarios), Japón (25.9 millones) y México (23.5 millones). [24]

Para ésta tesis se decidió obtener información en tiempo real, seleccionándose 3 cuentas de viajeros por considerarse de los más activos en sus cuentas de Twitter (MexDesconocido, TravelReportMX y WorldThruMyEyes), para establecer un corpus de por lo menos 3000 *tweets* de viajeros en sitios turísticos en México.

La información recopilada será procesada con el algoritmo SVM otras herramientas de utilidad que ayuden a generar un reporte final que permita conocer la diversidad de lugares a conocer en nuestro país, en la toma de decisiones de viajeros y permita mejores experiencias de viajes de acuerdo a gustos y necesidades; logrando tener un panorama más amplio de los lugares que se desean visitar y conocer en México.

1.6 Alcances y limitaciones

1.6.1 Alcances

Analizar *tweets* (*post* y *comments*) de al menos 3 cuentas diferentes de Twitter enfocadas al sector Turismo, auscultadas por un *listener* durante 4 meses como mínimo, recopilando la información en tiempo real en una base de datos NoSQL, que es la que permite almacenar información de éste tipo.

La información será procesada por un algoritmo de aprendizaje de máquina supervisado, para la clasificación de sentimientos y así, poder generar un proceso que ayude a la toma de decisiones sobre viajes en la república mexicana.

1.6.2 Limitaciones

La generación del proceso depende de los cometarios que realizan los usuarios en Twitter que no siempre son aportaciones dirigidas al tema de interés, por lo cual se espera obtener la mayor información de calidad posible que ayude a generar un proceso que funcione como ayuda en la elección de viajes.

1.7 Organización de la tesis

El presente documento de tesis se encuentra organizado de la siguiente forma:

En el Capítulo 1 se encuentran los antecedentes de la investigación, la justificación, el objetivo general, los objetivos particulares, justificación, beneficios esperados y los alcances y limitaciones del presente trabajo.

El Capítulo 2 hace mención a trabajos de investigación que han empleado Twitter para la recuperación de la información y la aplicación de algoritmos de *machine learning* para clasificación; así, como conceptos de los algoritmos y del proceso que se aplican en esta investigación. Se establece el Estado del Arte para esta investigación y se fundamenta este trabajo de tesis, con el Marco Teórico establecido para este trabajo.

El Capítulo 3 se explica la arquitectura del proceso propuesto, donde se puede visualizar por medio de diagramas el funcionamiento de la arquitectura propuesta para éste trabajo.

El Capítulo 4 se explica la implementación del proceso propuesto por medio de diagramas de flujos de datos.

El Capítulo 5 se muestran las pruebas y los resultados obtenidos para la generación de un reporte amigable para el usuario y que sirve de ayuda a viajeros.

2) Capítulo II. Marco Teórico y Estado del Arte

2.1 Introducción

En esta sección se establece el Estado del Arte para esta investigación y se describen trabajos relacionados con análisis y procesamiento de información en Twitter y la aplicación de diferentes algoritmos, para el procesamiento de información que sirven como base para fundamentar el presente trabajo.

2.2 Estado del arte

Existen diversos artículos, tesis de maestrías y referencias web que se han enfocado en el estudio y procesamiento de información a través de Twitter, las cuales las cuales se mencionan a continuación.

2.2.1 Detección de acoso en mensajes de Twitter

El trabajo presentado en el año 2017 por J.C. Ramos [1], publicado en EBSCO es uno de los temas más relevantes actualmente en México es el acoso a través de redes sociales; por lo cual se implementó un método de detección de acoso a través de Twitter, el cual recupera mensajes de Twitter con posibles palabras de acoso, pelea, exclusión, revelación, suplantación y los procesa, a través de ciertos algoritmos de aprendizaje automático mezclados en una técnica de ensamble, algoritmo de Naïve Bayes, regresión logística y máquinas de soporte vectorial. Para la clasificación se usó la herramienta Weka, con diferentes algoritmos de aprendizaje automático como: árboles predictores, árboles de decisión, K-vecinos más cercanos, basados en el teorema de Bayes y SMO (Sequential Minimal Optimization, Optimización mínima secuencial).

2.2.2 *Identificación de espacios de consenso en redes sociales basado en análisis semántico de producciones lingüísticas*

Debido a que Twitter no es una red social de uso exclusivo para personas físicas reales, las marcas aprovechan el medio de comunicación en tiempo real para ofrecer un canal de comunicación que muchos clientes prefieren por encima de otros medios. Twitter maneja un algoritmo de recomendación y aunque no está clara la naturaleza de mismo, algunos análisis permiten suponer la manera en la que funciona.

El trabajo presentado en el año 2014 por David Álvarez [2], publicado por EBSCO surge para identificar temas de intereses en común a partir de producciones lingüísticas entre usuarios en la red social Twitter, por lo que se construyó un corpus de producciones lingüísticas a partir de una muestra definida de usuarios de Twitter para identificar temas latentes usando *LDA* (Asignación de Dirichlet Latente, modelo probabilístico generativo basado en una distribución de distribuciones para colecciones de datos discrecionales como un corpus) y asignando un grado de pertenencia a los tópicos para cada uno de los usuarios, así como también se utilizaron algoritmos *tf·idf* (para saber qué tan relevante es una palabra en un documento dentro de una colección de documentos), *distancia euclidiana* (medida de distancia calculable entre dos puntos en un espacio euclídeo), *métricas de evaluación*, *precisión*, *recall* y *exactitud*.

2.2.3 *Desarrollo de un sistema de análisis de sentimiento sobre Twitter*

El trabajo presentado en el año 2015 por J. Selva [3], publicado por la Universidad Politécnica de Valencia, plantea un sistema de análisis de sentimientos sobre Twitter en español, debido a la ausencia de herramientas de este tipo, específicas para dicho idioma. Con este objetivo, este proyecto, parte del clasificador de polaridad desarrollado por Pla y Hurtado de la Universidad Politécnica de Valencia, para elaborar una aplicación web con Django que extrae estadísticas de polaridad de un tema concreto a petición del usuario. Se utiliza el clasificador SVM (Support Vector Machines) por haber obtenido los mejores resultados de asignación de polaridad sobre Twitter en español en el TASS2014 con una precisión del 62.88%.

Todo esto con el fin de extraer los *tweets* más relevantes de un conjunto de *tweets*. Para ello se considera el conjunto total de *tweets* como el documento a resumir y cada tweet una frase de este.

2.2.4 *Análisis de sentimientos y predicción de eventos en Twitter*

El trabajo presentado en el año 2014 por L. Montesinos [4], publicado por la Universidad de Chile; en la cual, se aplica análisis de sentimientos para la predicción a través de Twitter de cuánta es la participación de la población chilena en votos de elecciones, analizar usuarios con mayor número de *tweets* determinando las causas de esto, analizar usuarios con mayor cantidad de *tweets*.

Para determinar la polaridad (positivo, negativo, neutro) de los *tweets* se utilizó en lenguaje Perl, así como también un diccionario de palabras positivas y negativas. Se obtiene una opinión global por usuario, para saber si está a favor o en contra de algún candidato. Una vez obtenidos los resultados se comparan con los de las elecciones para determinar su eficacia.

Se ocuparon métodos de aprendizaje computacional (machine-learning approaches) y diccionarios léxicos.

2.2.5 *Sentiment classification of online reviews to travel destinations by supervised machine learning approaches*

El trabajo presentado en el 2009 por Qiang Ye, Ziqiong Zhang y Rob Law [5] publicado en *Elsevier*, es el que más se aproxima a la investigación planteada en este documento de tesis, ya que se basa en un estudio de destinos turísticos específicos basados en blogs. Utilizan 3 algoritmos de aprendizaje de máquina para clasificar la información como Naïve Bayes, SVM y el modelo de N-gramas. Donde los métodos de SVM y n-gramas dieron resultados más exactos, lo cual pudo realizar un mejor análisis sobre las preferencias de los destinos turísticos según los blogs analizados.

2.2.6 *Foundations of statistical natural language processing*

En 1999, Christopher. D. Manning y Hinrich Schütze publicaron en *Massachusetts Institute of Technology* [6] su libro que explica como el espacio vectorial se convierte en un modelo utilizado muy ampliamente en las ciencias de la computación. Su amplio uso se debe a la simplicidad del modelo y de su muy clara base conceptual que corresponde a la intuición humana en el procesamiento de información y de datos. Realmente la idea detrás del modelo es muy sencilla y es una respuesta a la pregunta ¿cómo pueden compararse los objetos de manera formal?

Parece que la única manera de describir los objetos es utilizar la representación con los rasgos (características) y sus valores. Es una idea universal, y hasta parece que es la única manera posible de trabajar con los objetos de manera formal. El modelo de espacio vectorial en el análisis de similitud entre textos fue creado para procesar ciertos cálculos que ayudan a entender al equipo de cómputo la similitud de texto que hay entre objetos. NLP es difícil por la ambigüedad del lenguaje.

2.2.7 Twitter Sentiment Analysis

En 2009, Alec Go, Lei Huang y Richa Bhayani, explican en su artículo [7], publicado por la universidad de Stanford, el análisis que debe de realizarse para que la computadora entienda sentimientos en un texto (positivos, neutrales, negativos) representados en un simple algoritmo de Orientación Semántica. En Twitter no existe un conjunto de datos de mensajes de sentimientos, por lo cual en éste artículo se creó un conjunto de datos para análisis de sentimientos. Se usaron clasificadores como el de Naive Bayes. Para descartar algunas inútiles características de nuestro texto, se utilizaron 3 diferentes algoritmos de selección de características: selección de características basada en frecuencias, información mutua y selección de característica x^2.

2.2.8 Lexicon-Based Sentiment Analysis of Twitter Messages in Spanish

El artículo escrito por Antonio Moreno-Ortiz y Chantal Pérez Hernández [8], publicado en 2013 por la facultad de Filosofía y Letras de la Universidad de Málaga, habla sobre los enfoques al análisis de sentimiento basados en lexicones difieren de los más usuales enfoques basados en aprendizaje de máquina en que se basan exclusivamente en recursos que almacenan la polaridad de las unidades léxicas, que podrán así ser identificadas en los textos y asignárseles una etiqueta de polaridad mediante la cual se realiza un cálculo que arroja una puntuación global del texto analizado. Estos sistemas han demostrado un rendimiento similar a los sistemas estadísticos, con la ventaja de no requerir un conjunto de datos de entrenamiento. Sin embargo, pueden no resultar ser óptimos cuando los textos de análisis son extremadamente cortos, tales como los generados en algunas redes sociales, como Twitter. En este trabajo se lleva a cabo tal evaluación de rendimiento con la herramienta *Sentitext*, un sistema de análisis de sentimiento del español.

2.2.9 *Sentiment Classification using Machine Learning Techniques*

En el artículo creado por Bo Pang, Lillian Lee y Shivakumar Vaithyanathan [9], publicado por Association for Computational Linguistics se realizan pruebas de clasificación de sentimientos (positivo y negativo) usando técnicas de *machine learning* (Naïve Bayes, SVM y clasificación de máxima entropía) en reseñas de películas como datos. Efectivamente se llega al resultado que usando métodos de *machine learning* se resuelven muchos problemas que se generaban con los resultados de humanos; donde SVM tiene mejores resultados de Naïve Bayes.

2.2.10 *Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases*

Éste articulo [10], publicado por International Journal of Computer Applications da una visión de cuando utilizar bases de datos relacionales y cuando NoSQL. Las bases de datos relacionales son ampliamente usadas en más aplicaciones de almacenamiento y recuperación de datos. Éstas trabajan mejor cuando manejan un conjunto de datos limitado. Manejando altos volúmenes de datos en tiempo real como internet fue ineficiente en sistemas de bases de datos relacionales. Para vencer éste problema comenzaron a existir las bases de datos NoSQL (No sólo SQL).

Además, muestra un cuadro comparativo de las bases de datos NoSQL, donde explica su rendimiento y características de cada una de ellas.

Existen 3 mayores problemas con las bases de datos relacionales:

1. El tamaño del conjunto de datos: Hay un alto crecimiento de información en internet, como se muestra en la Fig. 2.1 a continuación:

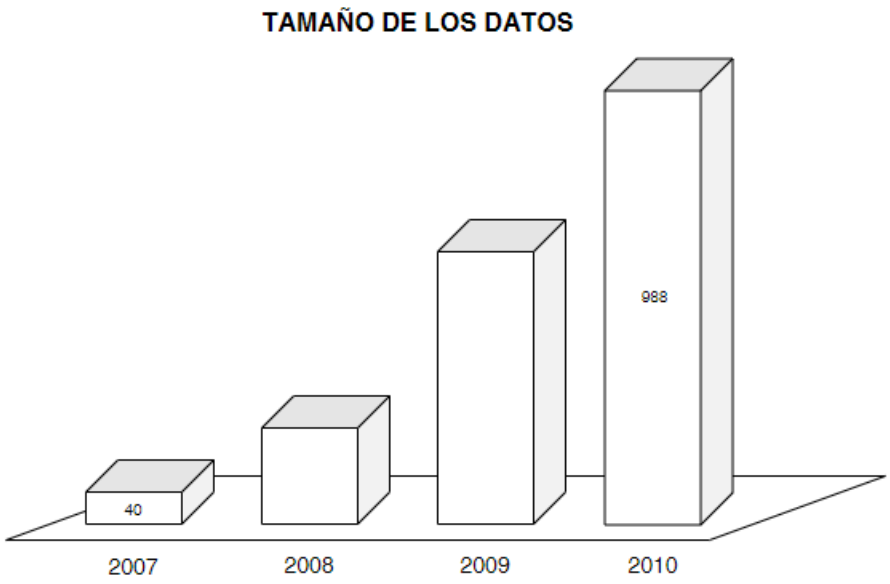


Fig. 2-1 Tamaño de datos de 2007 – 2010. [10]

2. Problema en conectividad. Conforme avanza el tiempo, la información está siendo más y más conectada. Como se muestra en la Fig. 2.2.

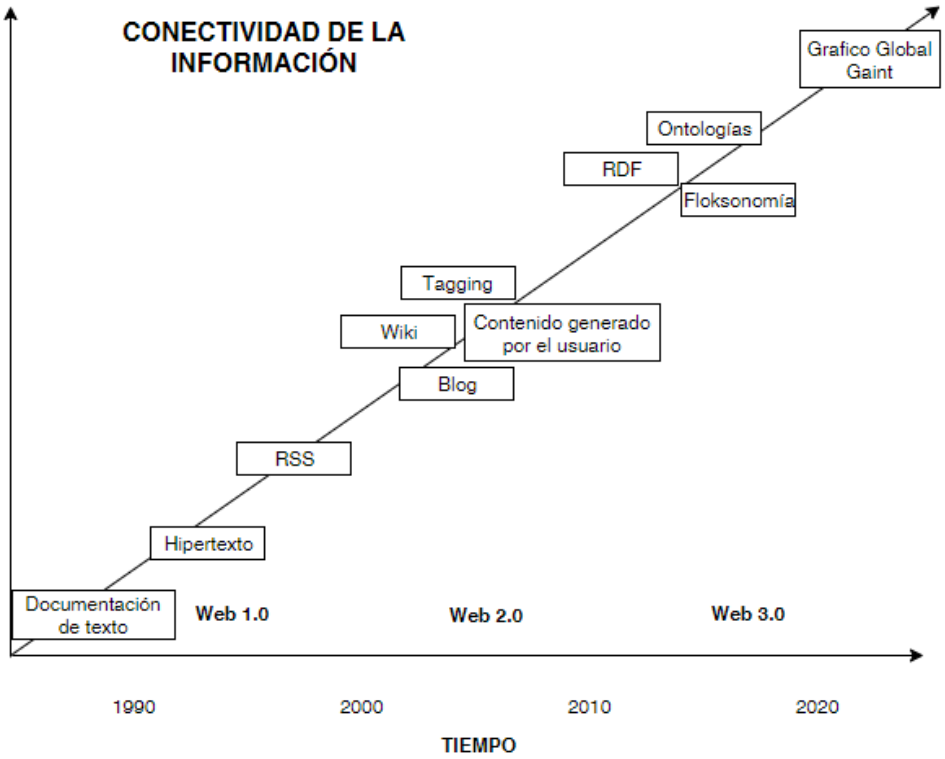


Fig. 2-2 Crecimiento de la conectividad. [10]

3. El tercer problema es acerca de la semi-estructura. La información semi-estructurada es información la cual tiene pocos atributos obligatorios pero muchos atributos opcionales. Como la información crece, hubo necesidad de incrementar columnas en la tabla, la cual suele conducir a escasas tablas.

2.2.11 Cuadro comparativo del estado del arte

En la Tabla 1, se muestra una tabla comparativa de los trabajos que se describieron en las secciones 2.2.1 a la 2.2.10 remarcando las similitudes en cuanto a la base de datos manejada, lenguajes de programación empleados, algoritmos utilizados, herramientas de lenguaje natural tomadas en cuenta, marco de trabajo (*framework*) utilizado y fuente de datos de la información (API).

Tabla 1 Cuadro comparativo del Estado del Arte. [Elaboración propia]

CUADRO COMPARATIVO DEL ESTADO DEL ARTE							
Tipo de trabajo	Nombre del Trabajo	Tecnología en uso					
		Base de Datos	Lenguaje de programación	Algoritmos	Natural Lenguage Tools	Framework	Fuente de Datos (API)
Tesis	Análisis de Sentimientos y Predicción de Eventos en Twitter	SQL	Perl	* SVM * Naive Bayes * Máxima entropía	corpus de emoticones, intensificadores.		Twitter
Tesis	Detección de acoso en mensajes de Twitter	Archivo de texto con codificación UTF-8	Python 2.7	* SVM	Stemming, lematización, stopwords, URLs, signos de puntuación, emojis, emoticones,	PyCharm	Twitter : (Streaming API)
Tesis	Identificación de espacios de consenso en redes sociales basado en análisis semántico de producciones lingüísticas	Redis	Rubi	* LDA * tf*idf * distancia euclidiana * metricas de evaluación		Sinatra	Twitter : (Search API)
Tesis	Desarrollo de un sistema de análisis de sentimiento sobre Twitter	MongoDB	Python 3.4	* LSA * SVM	Librerias: Tweepy, numPy, MatPlotLib, HTML, CSS,	Django	Twitter
Artículo	Sentiment classification of online reviews to travel destinations by supervised machine learning approaches	Páginas online acerca de los 7 destinos turísticos en el mundo		* Naive Bayes * SVM * N-gram			Twitter : (Search API)
Artículo	Twitter Sentiment Analysis	Corpus Edinburgh Twitter		* n-gram * lexicon features * micro-blogging features	tokenization, normalización, part-of-speech(POS) tagging, emoticons and abbreviations		Twitter
Artículo	Lexicon-Based Sentiment Analysis of Twitter Messages in Spanish	* Corpus de reseñas de peliculas * MySQL	C++	* SVM	Freeling	Sentitex	Twitter
Artículo	Sentiment Classification using Machine Learning Techniques	* Corpus de reseñas de peliculas		* Naive Bayes * Clasificación de máxima entropía * SVN * N-grams			Twitter

2.3 Marco Teórico

Hasta ahora se han empleado diversos algoritmos para el análisis de información a través de las computadoras, los cuales han ayudado a que la máquina no sólo vea los archivos como un simple medio de almacenamiento, sino que pueda interpretar y analizar la información que viene dentro de ellos. A continuación, se describen tecnologías empleadas, conceptos y artículos con información relacionada al tema de ésta tesis.

2.3.1 Stream API Twitter

Twitter Stream API es ideal para grabar datos en tiempo real y poder almacenar éstos para análisis. La librería más popular en Python para hacer streaming es *tweepy*, la cual permite conectar al Stream API y manejar errores propiamente.

Twitter Stream API es usado para descargar mensajes de Twitter en tiempo real. Esto es de ayuda para obtener un alto volumen de *tweets*, o para crear una transmisión en vivo usando un sitio de flujo o usuario de flujo.

Stream API es diferente de Search API. Search API muestra información muy similar a lo que se obtiene cuando se usa el buscador de Twitter, aunque está información se centra en la popularidad de los tweets o su cercanía temporal, más que en dar una información completa. Citando la documentación “the Search API is focused on relevance and not completeness” (Twitter Developers API – Search). Esto quiere decir que, con esta API, solamente tendré acceso a una pequeña parte del océano que es Twitter, aunque al menos se puede consolar si se piensa que será la más relevante. A diferencia de Search API, Stream API es de más utilidad en éste trabajo ya que se tiene acceso a la información completa. Funciona más como una “grabadora” en la que se van registrando todos los tweets a medida que se van creando. La ventaja es que no se te escapa nada, y la desventaja es que tienes que quedarte escuchando durante el tiempo de captura y no tendrías acceso a tweets antiguos [26].

En Tweepy, una instancia de tweepy.Stream establece una sesión de flujo y rutas de mensajes a la instancia StreamListener. El método on_data() de un stream listener recibe todos los mensajes y llamadas de funciones acorde el tipo de mensaje. El default StreamListener puede clasificar mensajes más comunes y rutas de ellos apropiadamente llamados métodos, pero esos métodos son solo de resguardo [11].

Por lo tanto, usando Stream API tiene 3 pasos:

1. Crear una clase heredando StreamListener.
2. Usando una clase que crea un objeto Stream.

3. Conectando a Twitter API usando el Stream.

Stream listener:

```
import tweepy
#override tweepy.StreamListener to add logic to on_status
class MyStreamListener(tweepy.StreamListener):

    def on_status(self, status):
        print(status.text)
```

En la Fig. 2-3, se muestra el diagrama de flujo del funcionamiento de la API [17].

Fig. 2-3Flujo de Stream API. [17, traducido al español]

2.3.2 CORPUS

Corpus o Corpora son esencialmente grandes colecciones de texto en forma electrónica, una colección especial de texto recolectado acorde a cierto conjunto de criterios. Están almacenados en computadoras y pueden ser manipulados con la ayuda de software conocido como herramientas de análisis de corpus. Corpora es considerado un buen recurso para gente interesada en estudiar idiomas, pero cabe aclarar que la forma en que las personas interactúan con los corpus es diferente a como lo hacen con los textos impresos. Por lo general, con los textos impresos, se consulta uno por uno y se leen en forma secuencial de inicio a fin. En contraste, cuando se estudia un corpus, generalmente se observan pequeños fragmentos de un

texto (por ejemplo, palabras individuales o líneas de texto individuales) y se pueden observar, múltiples fragmentos simultáneamente [12].

2.3.3 MongoDB

MongoDB es una base de datos NoSQL que ha surgido por la variación y complejidad en la información que se maneja actualmente en sistemas de cómputo. A diferencia de otras bases de datos NoSQL, MongoDB utiliza una colección de documentos que son interpretados por la maquina como objetos BSON, dichos objetos son muy similares a la estructura que manean los objetos JSON en Python; por lo cual, el manejo de información utilizando éstos objetos es muy compatible. [15]

El uso de MongoDB ha ido incrementando por el abastecimiento de principales servicios de negocios, sistemas de intranet y para construir análisis de *logs* y procesamiento de sistemas.

MongoDB y las bases de datos relacionales son diferentes en términos de implementación y conceptos operacionales. Alguna de la terminología usada para describir esto es mostrada en la Fig. 2.4.

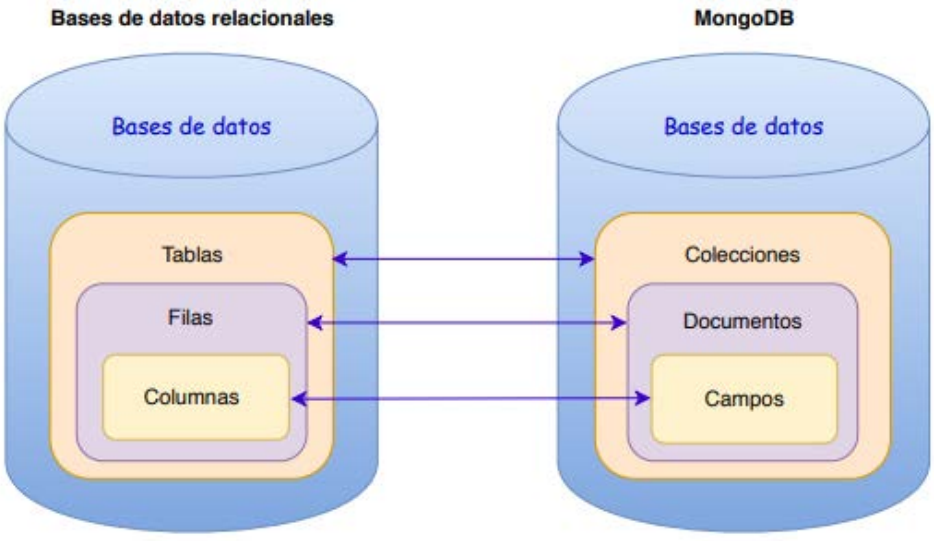


Fig. 2-4 Comparación de terminología de bases de datos. [15]

Como se observa en la Fig. 2-5 no hay un esquema en MongoDB y cada documento es almacenado en un formato de documento JSON. JSON es un formato estándar abierto para intercambio de datos ligeros basado en lenguaje JavaScript. En formato JSON, un objeto es un desordenado conjunto de pares de nombre/valor. Un objeto comienza con “{” y termina con “}”.

Cada nombre es seguido de ":" y los pares de nombre/valor son separados por coma ",". El valor puede contener un objeto y un arreglo. Un arreglo comienza con "[" y termina con "]". Documentos que en conjunto forman una colección de documentos.

Fig. 2-5 Ejemplo de una colección de documentos JSON. [Elaboración propia]

Las principales características de MongoDB es que son bases de datos flexibles en el almacenamiento, son rápidas y almacenan grandes volúmenes de datos como se muestra a continuación:

Velocidad: Las bases de datos documentales **son más rápidas**, atendiendo muchas operaciones por segundo. En la Fig. 2-6 se puede observar que gracias a la indexación las búsquedas en MongoDB son más rápidas y por su naturaleza permite búsquedas concurrentes.

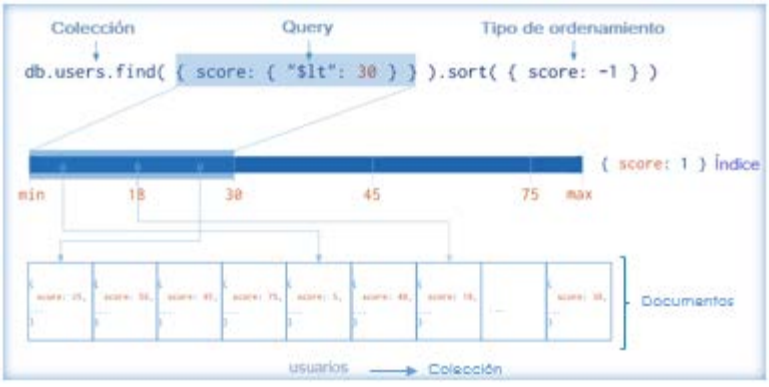


Fig. 2-6 Velocidad en bases de datos MongoDB. [Elaboración propia]

Volumen: Administran **grandes volúmenes de datos**. En la Fig. 2-7 se observa que MongoDB almacena sus colecciones de documentos en forma horizontal, lo cual hace tener acceso más fácil a la información y esto lleva a que cuando la información crece sólo se tienen costos de hardware.

Fig. 2-7 Volúmenes de datos en bases de datos MongoDB. [Elaboración propia]

Variabilidad: Cada documento puede almacenar campos distintos; es decir, **flexible en el almacenamiento**. En la Fig. 2-8 se muestran 2 documentos dentro de la misma colección de datos, que no necesariamente deben de tener la misma estructura de datos para funcionar de manera correcta. [27]

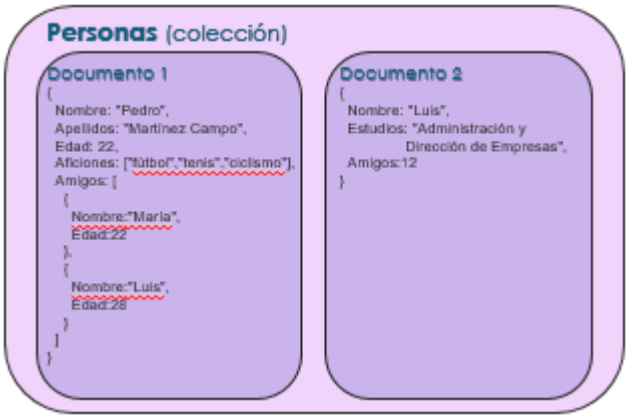


Fig. 2-8 Variabilidad en bases de datos MongoDB. [Elaboración propia]

2.3.3.1 Ranking de los sistemas de BBDD

En la Tabla 2 [14] obtenida de IEEE Conference Publications, se muestra un estudio que realizó DB-Engines en 2017, donde se observa el *ranking* del manejo de 339 bases de datos acorde a su popularidad.

Tabla 2 Ranking de Bases de Datos, 2017. [14]

339 systems in ranking, December 2017

Rank			DBMS	Database Model	Score		
Dec 2017	Nov 2017	Dec 2016			Dec 2017	Nov 2017	Dec 2016
1.	1.	1.	Oracle	Relational DBMS	1341.54	-18.51	-62.86
2.	2.	2.	MySQL	Relational DBMS	1318.07	-3.96	-56.34
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1172.48	-42.59	-54.17
4.	4.	4.	PostgreSQL	Relational DBMS	385.43	+5.51	+55.41
5.	5.	5.	MongoDB	Document store	330.77	+0.29	+2.09
6.	6.	6.	DB2	Relational DBMS	189.58	-4.48	+5.24
7.	7.	↑ 8.	Microsoft Access	Relational DBMS	125.88	-7.43	+1.18
8.	↑ 9.	↑ 9.	Redis	Key-value store	123.24	+2.05	+3.34
9.	↓ 8.	↓ 7.	Cassandra	Wide column store	123.21	-1.00	-11.07
10.	10.	↑ 11.	Elasticsearch	Search engine	119.78	+0.37	+16.51
11.	11.	↓ 10.	SQLite	Relational DBMS	115.19	+2.44	+4.36
12.	12.	12.	Teradata	Relational DBMS	74.74	-3.49	+1.37
13.	13.	↑ 14.	Solr	Search engine	66.30	-2.86	-2.70
14.	14.	↓ 13.	SAP Adaptive Server	Relational DBMS	65.68	-1.35	-4.74
15.	15.	↑ 16.	Splunk	Search engine	63.79	-1.08	+8.87
16.	16.	↓ 15.	HBase	Wide column store	63.41	-0.15	+4.79

Resulta significativo que en apenas 5-6 años de vida que tienen algunos motores NoSQL (No relacionales) se han llegado a **posicionar** hasta 3 sistemas, completamente diferentes en su modelo, **entre los 10 medios de almacenamiento más utilizados**. Seguramente sea debido a la evolución de las necesidades de ayer y de hoy. Las primeras BBDD (Bases de Datos) surgieron como una solución para el almacenamiento masivo de datos que se convirtió en necesidad para muchas empresas, conforme aumentaba el volumen de datos, éstas evolucionaron hasta surgir las BBDD Distribuidas, pero o bien porque no son una solución suficiente o por su complejidad, no fueron tan aceptadas como lo están siendo las soluciones NoSQL. Es posible concentrar las causas de la aceptación y popularización de los sistemas “No relacionales” en 3 grandes aspectos:

- Antes las bases de datos se diseñaban para ejecutarse en grandes y costosas máquinas aisladas. En cambio, hoy día, se opta por utilizar hardware más económico con una probabilidad de fallo predecible, y diseñar las aplicaciones para que manejen tales fallos que se consideran parte del “modo normal de operación”
- Los RDBMS (Relational Database Management System, Sistema de Gestión de Base de Datos Relacional) son adecuados para datos relacionados rígidamente estructurados, permitiendo consultas dinámicas utilizando un lenguaje sofisticado. Sin embargo, hoy

día, se desarrollan nuevas aplicaciones que se basan precisamente en datos con poca o ninguna estructura, dificultando su consulta por medios tradicionales.

2.3.3.2 Ventajas (Punto de vista NoSQL)

- **Evitar la complejidad innecesaria:** Los RDBMS proveen un conjunto amplio de características y obligan el cumplimiento de las propiedades ACID (Atomicity, Consistency, Isolation, Durability), sin embargo, para algunas aplicaciones (aquellas que necesiten una mayor disponibilidad y flexibilidad, en detrimento de la consistencia de los datos) éste conjunto podría ser excesivo y el cumplimiento estricto de las propiedades ACID innecesario.
- **Alto rendimiento:** Gracias a "sacrificar" la consistencia y centrarse en la disponibilidad de los datos puede conseguirse un mejor rendimiento.
  Ejemplo de rendimiento: una presentación realizada por los ingenieros Avinash Lakshman y Prashant Malik de Facebook, Cassandra puede escribir en un almacenamiento de datos más de 50 GB en solo 0.12 milisegundos, mientras que MySQL tardaría 300 milisegundos para la misma tarea.
- **Empleo de hardware más económico:** Las máquinas pueden ser mucho menos complejas (y baratas) y en caso de necesitar más potencia, pueden ser agregadas o quitadas sin el esfuerzo operacional que implica realizar *sharding* en soluciones de clúster de RDBMS.

Las NoSQL son diseñadas para almacenar estructuras de datos más simples o similares a las utilizadas en los lenguajes de programación orientados a objetos beneficiando principalmente a aplicaciones de baja complejidad.

- **El pensamiento "One-size-fits-all" estaba y sigue estando equivocado:** Existe un número creciente de escenarios que no pueden ser abarcados con un enfoque de base de datos tradicional.

En una entrevista a Ryan King, ex-Jefe Junior de Twitter encargado del uso de Cassandra: "Tenemos actualmente un sistema basado en MySQL compartido + memcached, pero se está convirtiendo rápidamente prohibitivamente costoso (en términos de recursos humanos) de operar. Necesitamos un sistema que pueda crecer de un modo más automatizado y tenga alta disponibilidad". [16]

MongoDB ofrece simplicidad, la curva de aprendizaje es corta para desarrolladores con experiencia en SQL y bases de datos tradicionales. Por otro lado, para ésta tesis es importante saber que se trabajará con *tweets* de Twitter, con los cuales es menos complicado, el manejo de

la extracción de ellos en objetos JSON (JavaScript Object Notation, Notación de Objetos de JavaScript), lo cual proporciona MongoDB [13].

2.3.4 *Análisis de Sentimientos*

Para identificar las opiniones en Internet, es necesario realizar un análisis de sentimientos, técnica que utiliza procesamiento de lenguaje, análisis de texto y herramientas computacionales para clasificar cometarios subjetivos de diferentes usuarios, ya sean sentimientos como tal u opiniones sobre diversos temas. Los métodos usados para este tipo de análisis tienen cerca de 15 años de aplicación y se han usado, para clasificar e-mails reseñas de clientes, publicaciones digitales, etc. A la hora de querer diseñar un sistema que analice y clasifique sentimientos u opiniones, hay que, en primer lugar, tener claro los desafíos que se deben vencer, los cuales están descritos en la literatura [18]:

- En primer lugar es necesario determinar si existe opinión en el *tweet* o no, ya que no siempre esto ocurre, pudiendo ser un comentario objetivo, una respuesta a otro usuario, etc.
- Determinar el tema sobre el cual se está hablando de manera de saber si es información útil, ya que se puede estar buscando opiniones sobre una empresa determinada y si el tweet es sobre política no aporta información relevante sobre lo que se está buscando.
- Reconocer las abreviaciones y modismos típicos. Al tener Twitter un carácter informal el lenguaje usado no siempre es correcto, ya que normalmente no se ocupan tildes y se ocupan palabras populares que no aparecen en el diccionario (Ej. Ocupar “bn” en vez de “bien”, “x” en vez de “por”, el uso de garabatos, usar expresiones del tipo “po”, ”maloooo”, etc.).
- Determinar la polaridad de una oración pudiendo tener palabras positivas y negativas en la misma frase (Ej. “Me alegro que se haya terminado, pésimo el espectáculo”, “La película no fue nada buena”). Los *tweets* a evaluar son todos aquellos que den una opinión, una evaluación o expresen emoción sobre algún tema de interés, dejando de lado los mensajes objetivos o informativos. Es así como existen varios procesos que se pueden aplicar para realizar un análisis de sentimientos en Twitter. En general, este tipo de problemas se resuelve catalogando una opinión en polaridades, determinando si es positiva o negativa con respecto a un tema específico. Sin embargo, este no es un tema simple de resolver, ya que dependiendo del contexto hay palabras que pueden expresar tanto una opinión positiva como negativa. En el caso de tener 2 polaridades, que es más usado en la literatura, cada mensaje puede ser catalogado como positivo o negativo. En éste proceso, se incluyen estudios sobre 8 tópicos distintos, como el caso de extraer opiniones en reseñas de películas o libros (“bueno” o “malo”), en opinión de productos (“me gusta” o “no me gusta”) o en elecciones políticas (“va a ganar”, “no va a ganar”). Además, se han considerado comúnmente 6 emociones universales: enojo,

disgusto, miedo, alegría, tristeza y sorpresa. De esta manera es posible catalogar los *tweets* de acuerdo a estas emociones de manera de determinar su polaridad y el grado de ésta, lo cual puede ser de gran utilidad para diferenciar mensajes en una mayor cantidad de categorías y no sólo positivo-negativo. De la misma forma, es posible hacer una escala gradual entre positivo-negativo, pudiendo tener 7 grados (3 positivos, 1 neutro y 3 negativos, variando de muy negativo a muy positivo) o 11 grados (-5 al 5, siendo -5 muy negativo, 0 neutro y +5 muy positivo). Para obtener la polaridad de un *tweet* existen 2 métodos que son los más usados. El primero de ellos, es usar algoritmos de aprendizaje computacional (machine-learning approaches) y el segundo, es utilizar diccionarios léxicos.

2.3.5 Aprendizaje Computacional

El aprendizaje computacional busca analizar la información de manera automática de una forma supervisada, basándose en *sets* de entrenamiento, los cuales serán usados para catalogar el resto de las opiniones encontradas en la Web, realizando pruebas y luego validándolas. Las principales técnicas de este método son: *Support Vector Machines (SVMs), Naive Bayes* y clasificadores de máxima entropía. De esta manera, se utiliza la categoría gramatical de las palabras, la presencia y frecuencia de algunos términos y su composición semántica.

2.3.6 Normalización

La normalización generalmente se refiere a una serie de tareas relacionadas destinadas a poner todo el texto en igualdad de condiciones: convirtiendo todo el texto en el mismo caso (superior o inferior), eliminando la puntuación, convirtiendo los números a sus equivalentes de palabras, y así sucesivamente. La normalización pone todas las palabras en pie de igualdad, y permite que el procesamiento proceda de manera uniforme.

La normalización del texto puede significar realizar una serie de tareas, pero para nuestro marco se aborda la normalización en 3 pasos distintos: derivación, lematización y todo lo demás (convertir todo el texto a minúsculas, eliminar puntuación, reemplazar números por texto y remover *stopwords)*.

2.3.7 Support Vector Machine (SVM)

SVM es un algoritmo de aprendizaje automático supervisado el cual puede ser usado por clasificación o regresión. Sin embargo, es más usado en problemas de clasificación. SVM

modela un lugar para crear un espacio de características el cual es un espacio vectorial de n-dimensiones, cada dimensión representa una característica de un objeto en particular. En la clasificación de documentos, cada característica es la importancia de una palabra en particular.

Las Máquinas de Vector Soporte se fundamentan en el *Maximal Margin Classifier*, que a su vez, se basa en el concepto de hiperplano. Un hiperplano se define como un subespacio plano (Aplicando aprendizaje supervisado, el algoritmo SVM entrena los datos y da como resultado un hiperplano óptimo el cuál clasifica nuevos ejemplos.). En un espacio de dos dimensiones, el hiperplano es un subespacio de 1 dimensión, es decir, una recta. En un espacio tridimensional, un hiperplano es un subespacio de dos dimensiones, un plano convencional. Para dimensiones n>3 no es intuitivo visualizar un hiperplano, pero el concepto de subespacio con n−1 dimensiones se mantiene.

En la Fig. 2.9 se observa un ejemplo del espacio vectorial del algoritmo SVM donde se grafica cada dato de elemento como un punto en un espacio n-dimensional (donde n es el número de características que se tiene), con el valor de cada característica siendo el valor de una coordenada en particular. Después, se realiza la clasificación para encontrar el hiperplano que diferencia muy bien a las 2 clases (como se muestra en la siguiente imagen). [21]

Los vectores de soporte son los puntos de ambas clases más cercanos al hiperplano. SVM es una frontera la cual segrega mejor a las 2 clases *(línea de hiperplano*):

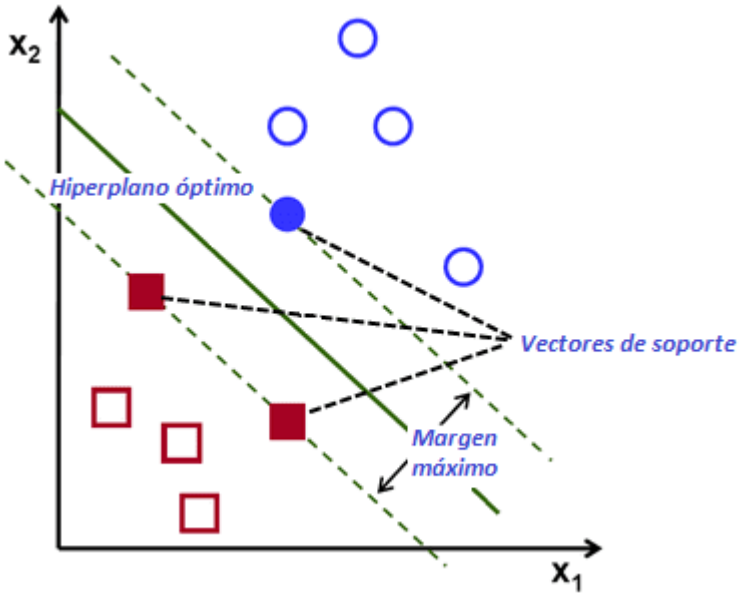


Fig. 2-9 Espacio vectorial de SVM en modelo lineal. [Elaboración propia]

2.3.8 *Modelo bolsa de palabras (Bag of Words)*

Con éste tipo de documento se simplifican los documentos a múltiples *sets* de frecuencias de términos. Significa para el modelo, una etiqueta de sentimientos en un documento va depender de qué palabras aparecen en ese documento, eliminando algunas gramáticas o palabras, pero guardando multiplicidad.

Un ejemplo de *Bolsa de palabras* se observa en la Fig. 2-10.

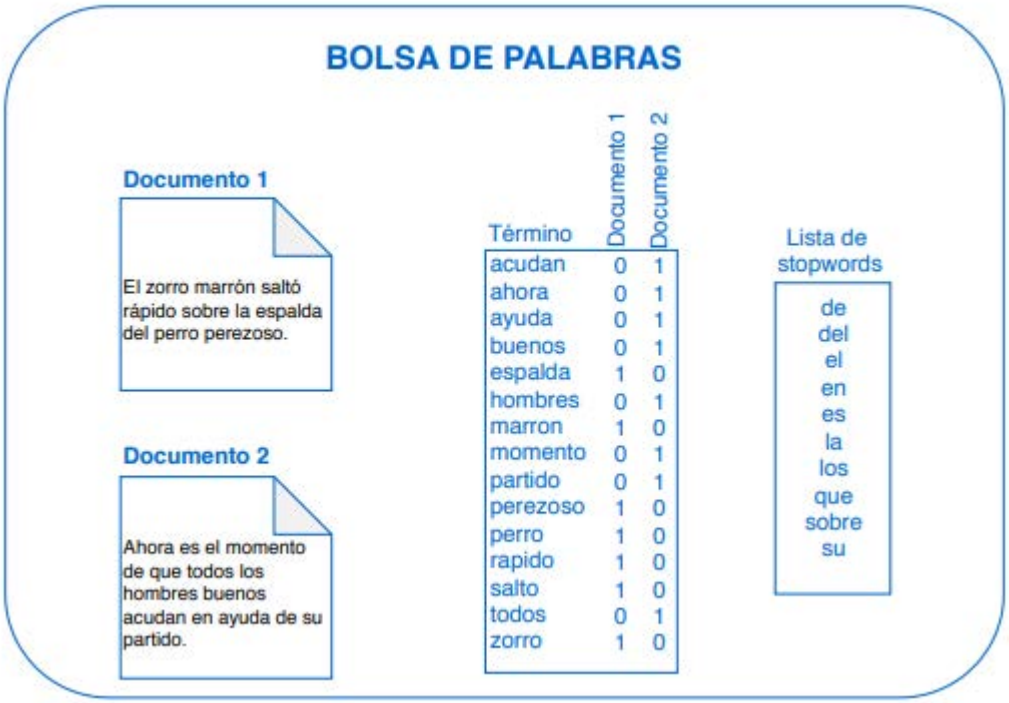


Fig. 2-10 Ejemplo de Bolsa de palabras.[Elaboración propia]

2.3.9 *tf*ifd*

El término tf·idf se refiere a la frecuencia de una palabra multiplicada por la frecuencia de aparición inversa de dicha palabra en una colección de documentos (la abreviatura del término proviene de sus siglas en inglés, *term frequency· inverse document frequency*. Se trata de una representación numérica de qué tan relevante es una palabra para un documento en específico dentro de una colección de documentos.

$$tf\Delta idf(t, d,D) = tf\ (t, d) \times idf(t,D)) \qquad (2.1)$$

El valor de **tf·idf** aumenta conforme un término aparece más veces en el documento en cuestión y menos en los otros documentos, aumentando la relevancia de dicho término para ese documento. Asimismo, el valor de esta métrica disminuye cuando el término se repite numerosas veces en todos los documentos, para palabras tales como artículos o preposiciones.

$$idf(t,D) = \log (N / |d \in D : t \in d|) \quad (2.2)$$

Esta medida se define en la ecuación 2.1, donde la función *tf* puede ser la frecuencia bruta del término *t* en el documento *d* y la función *idf* se encuentra en la ecuación 2.2.

2.3.10 Distancia Euclidiana

La distancia euclidiana es una medida ordinaria de distancia calculable entre dos puntos en un espacio euclídeo. En un espacio de dos dimensiones, esta medida corresponde directamente al teorema de Pitágoras, pero es aplicable también en espacios de tres o más dimensiones.

$$d_E(P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \quad (2.3)$$

La distancia euclidiana entre los puntos *P = (p1, p2, ..., qn)* y *Q = (q1, q2, ..., qn)* en un espacio euclídeo n-dimensional se define en la ecuación 2.3

2.3.11 Métricas de evaluación

Para evaluar el funcionamiento del sistema recomendador, se consideran los indicadores de *precision* y *recall*, típicos de tareas relacionadas con la clasificación de información. Ambas medidas hacen uso de los siguientes conceptos:

- **Verdadero positivo (vp).** Son los elementos que fueron etiquetados correctamente en la clase en turno.
- **Falso positivo (fp).** Elementos que fueron etiquetados en la clase en turno, pero no pertenecen a ella. Es decir, fueron etiquetados erróneamente como pertenecientes a la clase.
- **Falso negativo (fn).** Elementos que no fueron etiquetados en la clase en turno, pero sí pertenecían a ella.

2.3.11.1. Precision

$$precision = (vp / (vp + fp)) \quad (2.4)$$

La precisión, también conocida como:"valor predictivo positivo"; consiste en el porcentaje de predicciones extraídas que fueron catalogadas correctamente.

Se define como el cociente de los verdaderos positivos y la suma de los verdaderos positivos y los falsos positivos. Está representada en la ecuación 2.4

2.3.11.2. *Recall*

$$recall = (vp / (vp + fn)) \quad (2.5)$$

La medida de recuperación, o comúnmente conocida por su término en inglés: recall, es complementaria a la precisión y se refiere al porcentaje de las predicciones positivas que fueron extraídas del total de predicciones disponibles por extraer. Es decir, del total de predicciones o clasificaciones que debieron ser identificadas, cuántas en realidad fueron obtenidas.

Se define como el cociente de los verdaderos positivos y la suma de los verdaderos positivos y los falsos negativos. Está representada en la ecuación 2.5.

2.3.11.3. *Exactitud*

$$exactitud = ((vp + vn) / (vp + vn + fp + fn)) \quad (2.6)$$

La medida de exactitud, conocida también como *accuracy* en inglés, da una idea más general del total de clasificaciones realizadas, ya sean positivas o negativas, mientras sean verdaderas. Es decir, expone de una forma más directa qué tan exacto es el sistema al momento de etiquetar tanto términos positivos como negativos.

Se define como el cociente de la suma de los verdaderos positivos y los verdaderos negativos y la suma de los verdaderos positivos, los verdaderos negativos, los falsos positivos y los falsos negativos. Está representada en la ecuación 2.6.

2.3.12 Librerías para Python

2.3.12.1. *Tweepy*

Tweepy es una librería de código abierto para Python que incluye todo el conjunto de funciones necesarias para comunicar con Twitter mediante las API definidas por este. Las funciones definidas por Tweepy simplifican sobremanera la conexión y búsquedas con Twitter. Por ejemplo, toda conexión a Twitter debe estar certificada con OAuth y mientras que, por defecto, habría que configurar esta conexión mediante otra librería como Python-OAuth y establecer cada conexión manualmente, Tweepy simplifica esto con funciones que simplemente esperan los parámetros para configurar todo automáticamente. En el caso de OAuth, simplemente hay que pasarle los 4 tokens necesarios y en el caso de la búsqueda solo tienen que indicarse los parámetros que

solicita Twitter, toda la complejidad de las conexiones la trata internamente simplificando el trabajo inmensamente. Para el proyecto se ha empleado la versión 3.5.0 de Tweepy.

2.3.12.2. NumPy

Esta librería es una extensión de Python de código abierto que añade soporte para trabajar con grandes arreglos multidimensionales. Incluye la implementación de diversas operaciones matriciales de forma automática. *NumPy* permite realizar toda clase de operaciones complejas con matrices de manera eficiente y sencilla, por ello se convierte en la librería ideal siempre que haya que tratar con grandes cantidades de información en forma de matriz. Esta librería está destinada a un nivel inferior a Python, es decir sus funciones están dirigidas a trabajar sobre *CPython*, con una implementación que trata de sacar el máximo partido a las librerías en C de Python, buscando la máxima eficiencia.

Para este trabajo de tesis, tanto para el clasificador como para el sistema de resúmenes automáticos ha demostrado ser una librería de gran ayuda. En el caso del sistema de resumen, todas las matrices con las que trabaja *scikit-learn* (Subsección 2.6.3) están representadas como matrices de *NumPy*. Una función muy útil que incluye esta librería y que es primordial para el correcto funcionamiento del resumidor automático es la descomposición lineal SVD (Singular Value Descomposition), la cual ha simplificado la implementación de los resúmenes por ser una operación matricial bastante compleja. Para el proyecto se ha empleado la versión 1.11.3 de *NumPy*.

2.3.12.3. Scikit-Learn

Es una librería de código abierto para Python que añade funciones de aprendizaje automático. Implementa algoritmos de clasificación, regresión y *clustering* incluyendo algoritmos de Vectores de Máquinas Soporte o SVM (Support Vector Machines) que se emplean para el clasificador de polaridad.

Esta librería se ha empleado para crear automáticamente vectores de características (como se observará en el Capítulo 5), es decir, dado un conjunto de documentos, genera automáticamente matrices con características de las palabras en los documentos, agilizando mucho el tratamiento de los *tweets* a la hora de realizar los resúmenes. Para este proyecto se ha empleado la versión 0.19.0 de *Scikit-Learn*.

2.3.12.4. *BeautifulSoup*

Es una técnica llamada "*web scraping*" que se ocupa para obtener datos de páginas web en un formato que se pueda trabajar para un análisis.

2.3.13 BDpedia

DBpedia es un conjunto de datos que pueden ser accesados online vía un *endpoint* de consulta SPARQL y como datos vinculados.

El acceso de datos DBpedia permite posibilidades de repuesta bastante asombrosas contra datos Wikipedia. Hay un *endpoint* SPARQL público por encima del conjunto de datos DBpedia en http://dbpedia.org/sparql. El *endpoint* es proporcionado usando OpenLink Virtuoso como el back-end del motor de la base de datos y el servidor HTTP.

El proyecto DBpedia ha generado durante mucho tiempo información semántica. Desde 2011 el proceso de generación de información extrae información de Wikipedia en sus 15 idiomas. El comité internacional de DBpedia ha asignado un sitio web y un SPARQL Endpoint para cada uno de estos idiomas.

La información completa sobre la sección del idioma español de la DBpedia (SPARQL EndPoint, datos, información para desarrolladores, etc.) se puede encontrar en el Wiki.

Ejemplo:

```
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?person WHERE{
   ?person dcterms:subject <http://es.dbpedia.org/resource/Categoría:Científicos_de_España>
}
```

Resultado:

Person
http://es.dbpedia.org/resource/Gustavo_Pittaluga_Fattorini
http://es.dbpedia.org/resource/Eduardo_Zorita
http://es.dbpedia.org/resource/Bernardo_Rodríguez_Largo
http://es.dbpedia.org/resource/Fermín_Muñoz_Urra
http://es.dbpedia.org/resource/Manuel_Lugue_Otero
http://es.dbpedia.org/resource/Mateo_Valero_Cortés
http://es.dbpedia.org/resource/Miguel_Ángel_Mayer
http://es.dbpedia.org/resource/Bernardino_Landete_Aragó
http://es.dbpedia.org/resource/Salvador_Gil_Vernet
http://es.dbpedia.org/resource/Gonzalo_Gallas_Novás
http://es.dbpedia.org/resource/María_Luisa_Ferrándiz_Manglano

http://es.dbpedia.org/resource/María_Teresa_Miras_Portugal
http://es.dbpedia.org/resource/Montse_Calleja
http://es.dbpedia.org/resource/Anastasio_Anselmo_González_y_Fernández
http://es.dbpedia.org/resource/Gabriel_Galán_Ruiz
http://es.dbpedia.org/resource/Amador_Schüller
http://es.dbpedia.org/resource/Pedro_Mayoral_Carpintero
http://es.dbpedia.org/resource/Guillermo_Velarde
http://es.dbpedia.org/resource/Tomás_Manuel_Vilanova_Muñoz_y_Poyanos
http://es.dbpedia.org/resource/Nuria_Martí_Gutiérrez
http://es.dbpedia.org/resource/Álvaro_Arias

Los datos vinculados (que será el acceso a datos usado en éste trabajo) usan información en la web para conectar datos relacionados que no fueron previamente vinculados, o usados en la web para reducir las barreras para vincular los datos actualmente vinculados mediante otros métodos. Más específicamente, Wikipedia define datos vinculados como “un término usado para describir a mejor practica recomendada para exponer, compartir y conectar piezas de datos, información, y conocimiento en la web semántica mediante URIs y RDF.”

En la Fig. 2-11 se muestra una imagen del diagrama del proyecto en la nube de datos abiertos conectados o vinculados; el cual existe para proporcionar un hogar o punteros a recursos de toda la comunidad de datos vinculados.

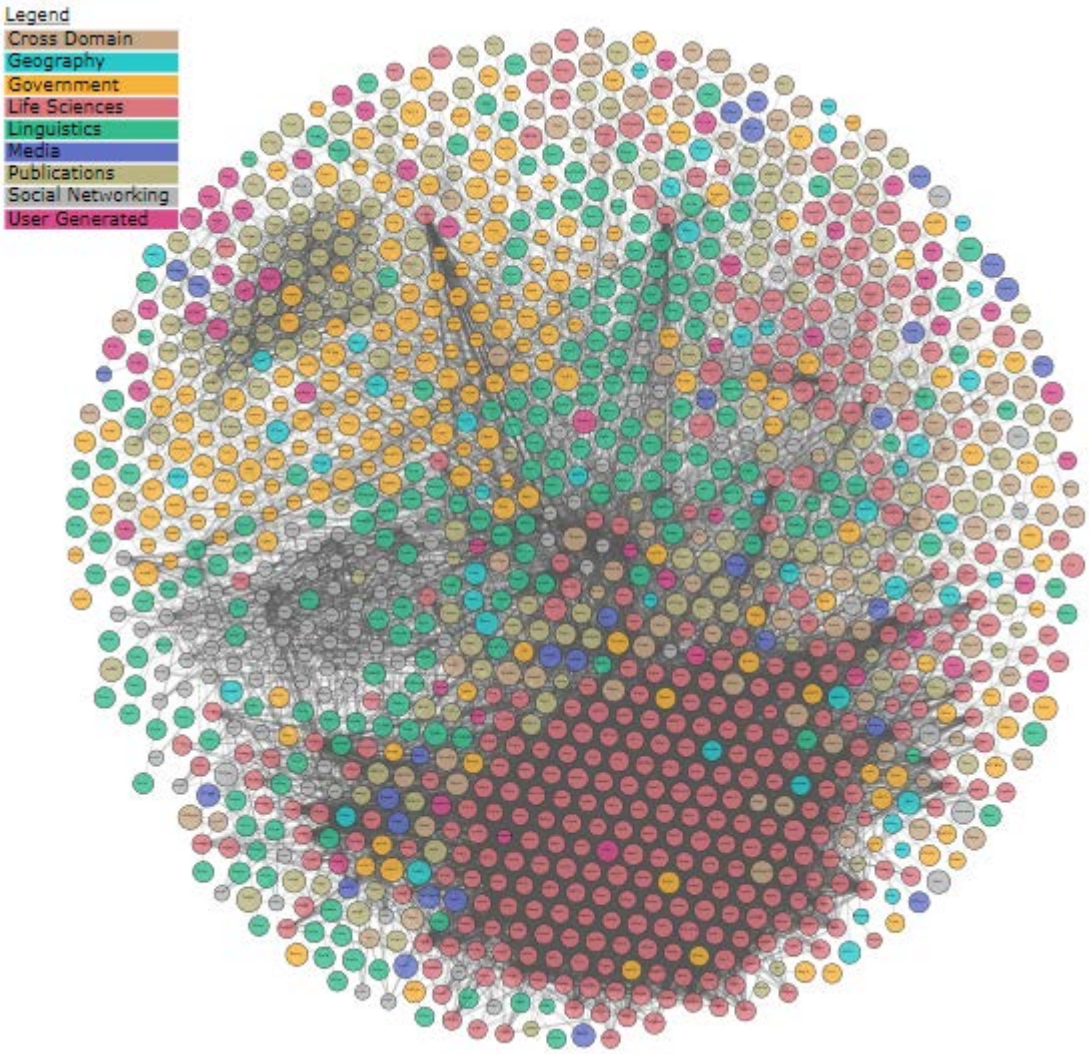


Fig. 2-11 Diagrama de proyecto de datos abiertos vinculados.[28]

2.3.14 Flask

Flask es un micro *framework* para Python basado en Werkzeug, Jinja 2, concebido para facilitar el desarrollo de aplicaciones Web bajo el patrón MVC. Permite crear de una manera muy sencilla aplicaciones web con Python como se muestra a continuación:

```
from flask import Flask
app = Flask(__name__)
@app.route("/")
def hello():
return "Hello World!"
```

Flask depende del motor de template Jinja y de la herramienta WSGI Werkzeug. Es clasificado como un microframework porque no requiere particulares herramientas o librerías. No cuenta con una capa de abstracción de base de datos, validación de *form* o algún otro componente donde preexistan librerías tripartitas provistas por funciones en común.
Debido a que se cuenta con un diccionario de datos con la información a mostrar en la página web, se eligió éste microframework para la generación de la página web que mostrará con mayor claridad y mejor manejo las recomendaciones de viajeros obtenidas de Twitter. [25]

2.3.15 Resumen del capítulo

En éste capítulo se estudió, analizó y revisaron las diferentes herramientas y algoritmos que de acuerdo en la investigación que se hizo en el estado del arte, ayudarán al desarrollo de éste proyecto. Con lo descritos en las secciones anteriores, se pretende dar las bases para comprender mejor el funcionamiento y utilidad de cada una de las herramientas y algoritmos mencionados que serán empleados en este trabajo de tesis para el análisis y visualización de resultados.

3) CAPITULO III Proceso propuesto: Reporte de Viajeros MX

3.1 INTRODUCCION

En éste capítulo se describe la lógica de los algoritmos empleados, así también se presenta y describe uno de elaboración propia, con los cuales, se genera el proceso propuesto en forma de diagramas de flujos, para finalmente mostrar un reporte con la información más útil, seccionada y de fácil acceso para que éste reporte sirva como base para futuros viajeros.

3.2 ARQUITECTURA DEL SISTEMA

Para ésta tesis se propone la generación de un reporte para viajeros que deseen visitar México, él cuál muestra recomendaciones y no recomendaciones por parte de las experiencias que han tenido otros usuarios viajeros y lo dan a conocer en Twitter. La arquitectura para la generación de éste reporte hace uso de diversas herramientas de software que ayudan a manipular la información (tweets) y así mostrar la información más útil, de una manera organizada y clara (véase Fig. 3-1).

Fig. 3-1 Proceso: Reporte de Viajeros MX [Elaboración propia]

En las siguientes secciones, se explica cada paso de la arquitectura.

3.2.1. Elección de cuentas de Twitter

Inicialmente se estudiaron y eligieron 3 cuentas de diferentes usuarios viajeros de Twitter, se toman estas cuentas debido a mostraron características de ser cuentas activas y de las más usadas por usuarios viajeros.

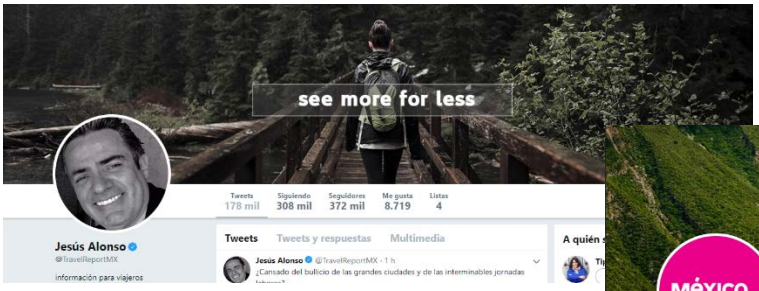


Fig. 3-3 @TravelReportMX

Fig. 3-2 @mexdesconocido

Fig. 3-4 @WorldThruMyEyes

3.2.2. Generación de Corpus

Se desarrolló un algoritmo StreamListener que se encarga de escuchar cada tweet de las 3 cuentas de Twitter en tiempo real durante un periodo de 6 meses, éste algoritmo hace uso del API de Twitter Streaming; el cuál, realiza un llamado al servidor Linux MongoDB y con la llegada de cada evento, inserta cada tweet en la base de datos. El flujo de éste algoritmo es como se muestra en la Fig. 3-5.

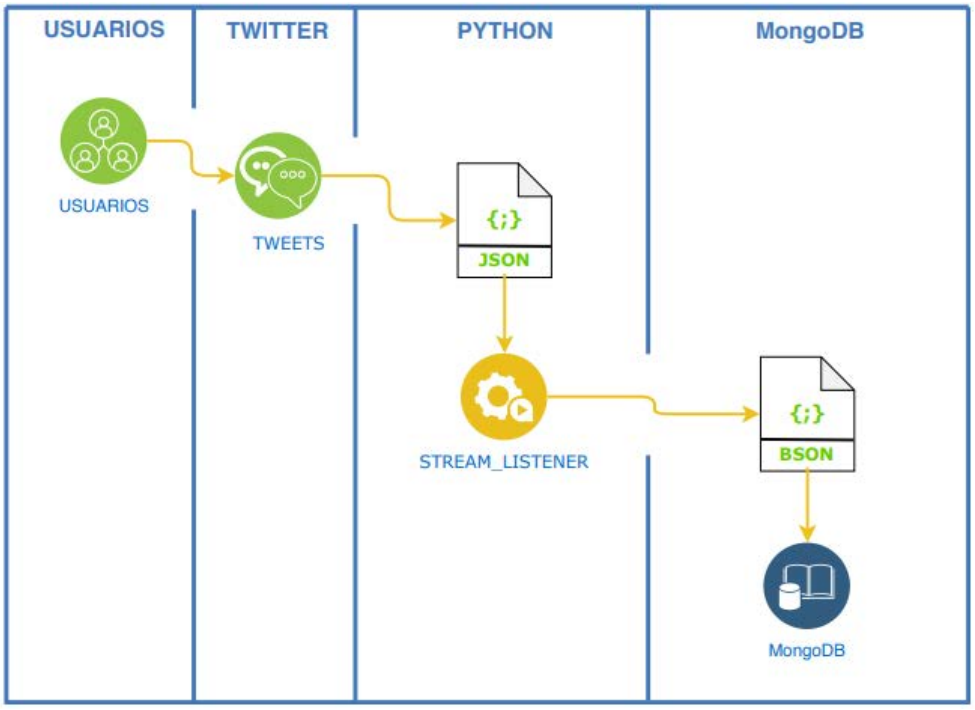


Fig. 3-5 Arquitectura de generación de corpus [Elaboración propia].

3.2.3. *Clasificar Tweets*

Se genera un algoritmo que se encarga de obtener el corpus de MongoDB y organiza la información por ID para mostrarla como se observa en el *timeline* de cada usuario de Twitter. Con la ayuda de algoritmos de lenguaje natural se aplican los siguientes estándares para normalizar (limpiar) el corpus (*tweets*):

- Eliminación de ruido (corchetes, brakets, tags de objeto JSON)
- Convertir todo el texto en minúsculas
- Cambiar números por texto
- Eliminar las URL
- Eliminar puntuación (‘.’,’:’,’…’,’,’,’;’, ‘@’,’#’ etc.)
- Eliminar *stopwords*
- Lematización de palabras

Posteriormente, se genera un Excel con todos los *tweets* obtenidos y se aplica el *aprendizaje supervisado* para clasificar cada tweet como: Nada recomendado (-2), No tan recomendado (-1), neutro (0), Recomendado (1), Muy recomendado (2). Finalmente se aplica el algoritmo Support Vector Machine (SVM) para clasificar linealmente el modelo. Si el hiperplano del modelo es mayor a 0.8, se entiende que la clasificación es buena y se procede a generar el “Reporte de Viajeros MX” (Véase Fig. 3-6).

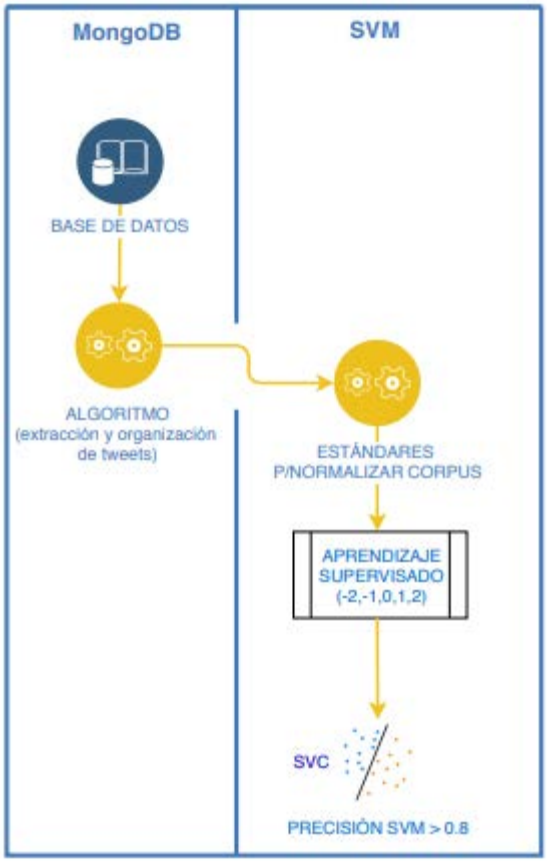


Fig. 3-6 Flujo de clasificación de tweets. [Elaboración propia]

3.2.4. Generar "Reporte de Viajeros MX"

Para mostrar los *tweets* de una forma clara a los viajeros, se configuró un ambiente Linux para levantar un servicio de base de datos local DBpedia, en el cuál por medio de un algoritmo que se generó, se ejecutan *queries* con cada uno de los *tweets* y DBpedia regresa un objeto XML en el cual se determina si el tweet habla de un estado de México, otro lugar, comida u otros. De ésta manera se puede filtrar la información de una manera comprensible al usuario y por tipo de recomendación. Se configuró un ambiente con el micro framework Flask para mostrar los *tweets* de una forma ordenada y dinámica en una página web y así poder observar las recomendaciones y no recomendaciones de los usuarios viajeros por medio de un "Reporte de Viajeros MX".

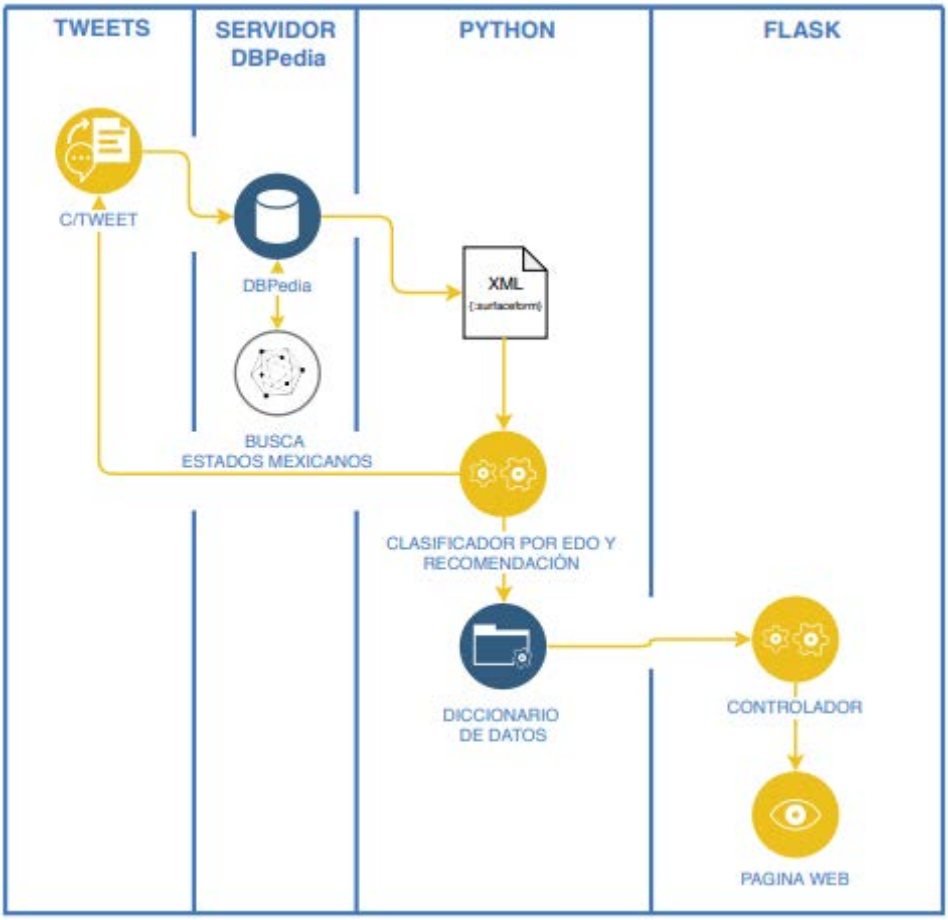


Fig. 3-7 Arq. Reporte para Viajeros MX. [Elaboración propia]

4) Capítulo IV. IMPLEMENTACION

4.1 Extracción y guardado de la información de Twitter desde servidor Linux

En la obtención de información es de ayuda la librería **tweepy** y el objeto **StreamListener** (StreamListener es una de las API de Twitter) para obtener *tweets* en tiempo real (ya que dicho objeto se encuentra escuchando todo el tiempo y captura cada tweet cuando existe algún evento en las cuentas elegidas de Twitter (*@TravelReportMX, @MexDesconocido, @WorldThruMyEyes*), como lo son: *posts, comments, reply*, etc.).

Una vez que se escucha un evento en la cuenta elegida, para guardarlo en *MongoDB* se ocupa el objeto **MongoClient** de la librería **pymongo**, el cual hace inserciones en la base de datos No SQL por cada evento que se escucha en tiempo real. Esto genera para una máquina de trabajo mucho desgaste debido a que debe estar prendida y escribiendo en disco duro por 24 horas durante al menos 4 meses (tiempo aproximado que se calculó para obtener un corpus decente), por lo cual se utilizó un servidor remoto Linux en el cual se mantuvo el proceso vivo hasta que se obtuvo el corpus deseado.

Para poder observar la información guardada en MongoDB y el formato en el que es guardada ésta información, véanse *figuras 4-1 y 4-2*:

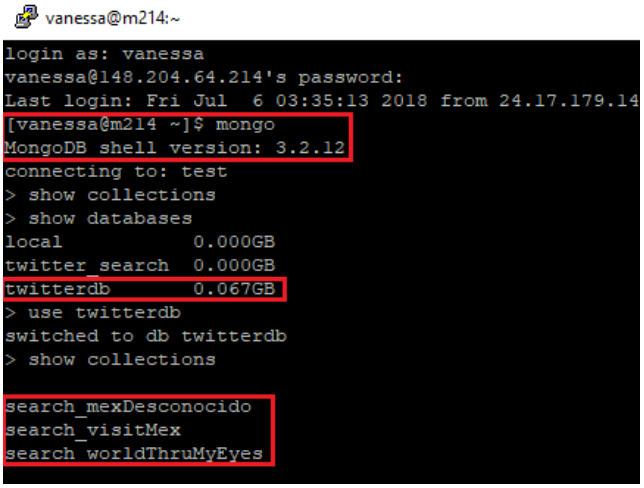


Fig. 4-1 Acceso a MongoDB [Elaboración propia]

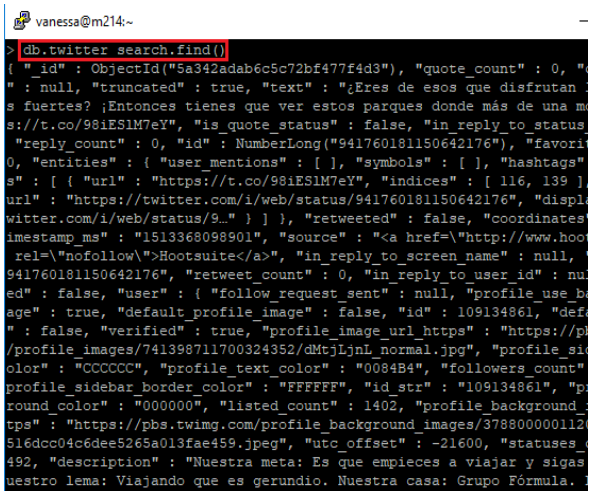


Fig. 4-2 Base de datos MongoDB [Elaboración propia]

Ésta información es guardada como objetos BSON (Binary JSON, JSON Binario) que son objetos parecidos a objetos JSON (objetos usados en Python) considerados objetos "más" legibles, con información estructurada y de manejable acceso para Python. Debido al rating de uso de MongoDB en base de datos NoSQL, el manejo de información en objetos BSON y su fácil manejo fueron motivos de la elección de ésta base de datos NoSQL.

Objeto BSON

```
{
    "_id" : ObjectId("59e0d6be251497309421d136"),
    "quote_count" : 0,
    "contributors" : null,
    "truncated" : false,
    "text" : "#Patzcuaro tiene muchas historias por contarte #ViajemosTodosPorMéxico #PueblosMágicos https://t.co/YX8uHeXmNE",
    "is_quote_status" : false,
    "in_reply_to_status_id" : null,
    "reply_count" : 0,
    "id" : NumberLong("918855763405107200"),
    "favorite_count" : 0,
    "entities" : {
        "user_mentions" : [ ],
        ...
    },
    ...
    "retweet_count" : 0,
    "in_reply_to_user_id" : null,
    "favorited" : false,
    "user" : {
        ...
    },
    "geo" : null,
    "in_reply_to_user_id_str" : null,
    "possibly_sensitive" : false,
    "lang" : "es",
    "created_at" : "Fri Oct 13 15:07:40 +0000 2017",
    "filter_level" : "low",
    "in_reply_to_status_id_str" : null,
    "place" : null,
    ...
    ...
    ...
}
```

Fig. 4-3 Objeto BSON [Elaboración propia]

El objeto StreamListener es un *listener* que realiza una conexión con cada uno de los usuarios de Twitter y se encarga de escuchar, obtener y guardar (en MongoDB) cada evento (tweet, retweet) que surge en cualquier comento (a esto se le llama que escucha en tiempo real), como se muestra en la Fig. 4-4.

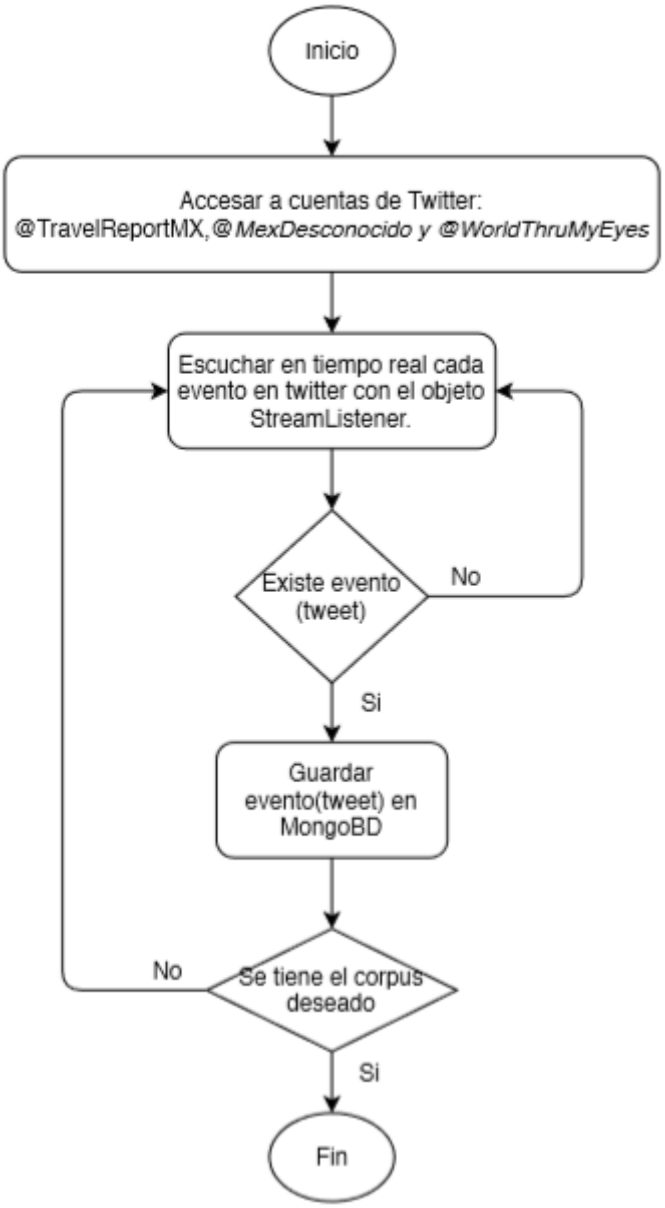


Fig. 4-4 Objeto StreamListener. [Elaboración propia]

4.2 Extracción y normalización de información (*tweets*)

La información guardada en MongoDB se extrae para iniciar con la normalización y el análisis de *tweets*. Para esto se eliminaron todos los *retweets* guardados en la base de datos ya que no proporcionaban información de ayuda y a su vez se generó un algoritmo en Python que conjunta y guarda en un archivo EXCEL cada *post* con sus respectivos *comentarios*, si es que dicho *post* tuvo *comentarios*, para su posterior análisis y normalización.

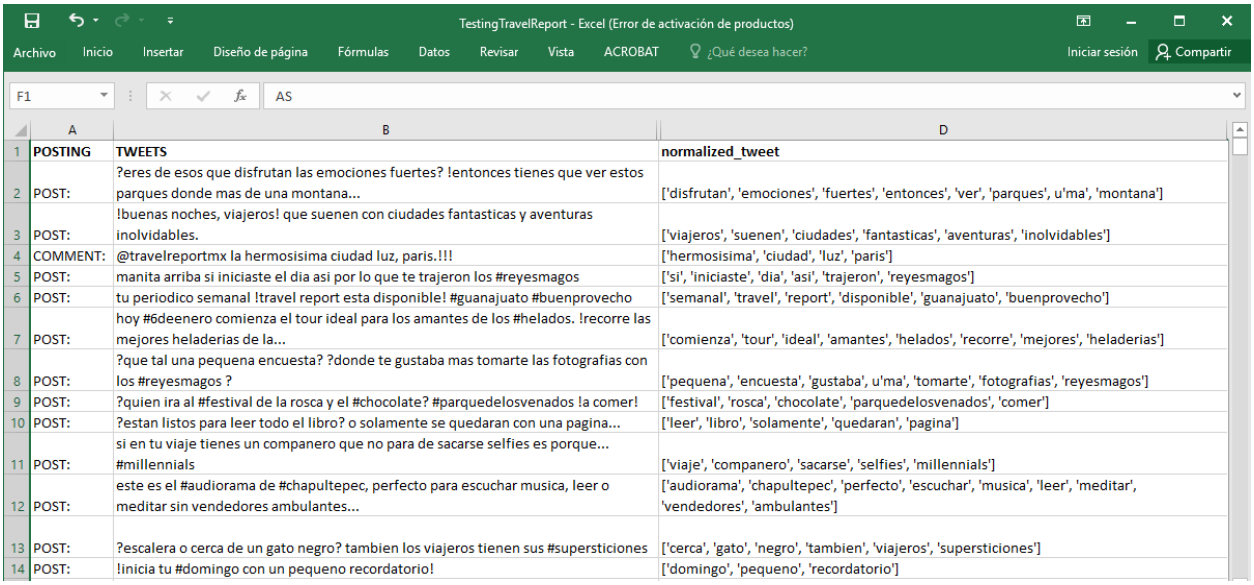
Para limpiar la información obtenida (*tweets*), se eliminó el ruido para obtener sólo el tweet (información que se extrae en objeto JSON), se normalizó la información de forma que todos los *tweets* fueran puestos en igualdad de condiciones, eliminando puntuación, caracteres HTML, *stopwords*, cambiando números por texto y se usó un corpus de *emoticones* para reemplazar éstos por texto [19]. (Véase la Tabla 3).

Tabla 3 Corpus Emoticones. [19]

EMOTICONES			
|:) estoy_feliz	u.u Cara_aflijida	n-n Carita_amable	^w^ Sonrisa_amistosa
:-) estoy_feliz	owo Cara_sonriente_ojos_abiertos.	^-^ Sonrisa_amable	:L Cara_de_tonto
:) estoy_feliz	:B Sonrisa_tonta_c_dientes_fuera.	^.^ Sonrisa_amable	:3 Cara_feliz
(: estoy_feliz	T-T Llanto	^^ Sonrisa_amable	¡-¡ Sorprenderse
(-: estoy_feliz	x3 Sonrisa_divertida_ojos_cerrados.	:v Cara_de_burla_o_tonta	@-@ Confundirse
:3 estoy_feliz	XD Cara_divertida	:,v Cara_de_burla_con_nariz	¬¬ Cara_enojada
:(estoy_triste	e-e Mirada_de_sospecha	>:v Cara_de_burla_enojada	*¬* Babear
:-(estoy_triste	7-7 Mirada_acusadora	:U Cara_de_tonto	y.y Llorar
): estoy_triste	7u7 Mirada_acusadora_picarona	>:c Carita_enojada	(9ò.ó)9 En_posición_de_ataque
)-: estoy_triste	77 Mirada_acusadora	._. Cara_de_poker	(/u\) Cubrirse_los_ojos_con_timidez
:') estoy_llorando	>-< Carita_angustiada	.-. Cara_de_póker_al_revés	(?) Cuando_una_oración_no_tiene_sentido
:O estoy_asombrado	>_< Carita_angustiada	-.- Cara_de_poco_interés	(͡° ͜ʖ ͡°) Carita_seductora
:d estoy_sorprendida	<_< Mirada_acusadora	_=3= Hacer_Pucheros	
:S estoy_confundido	<-< Mirada_acusadora	_=w= Relajado	
uwu Cara_sonriente_picara.	>///< Carita_apenada	0.o Impresionarse	

En la Fig. 4-5 se visualiza un ejemplo de la cuenta TravelReportMX, donde se puede apreciar una parte el resultado de normalización de *tweets*; donde la columna *TWEETS* refleja los tweets extraídos de

MongoDB organizados (post con comments) como muestra la columna *POSTING* y finalmente la columna *normallized_tweet* muestra los tweets normalizados. A éste corpus se le aplicará posteriormente el Aprendizaje Supervisado (AS) para poder entrenar el algoritmo SVM.



TestingTravelReport - Excel (Error de activación de productos)

Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista ACROBAT ¿Qué desea hacer? Iniciar sesión Compartir

F1 AS

	A	B	D
1	**POSTING**	**TWEETS**	**normalized_tweet**
2	POST:	?eres de esos que disfrutan las emociones fuertes? !entonces tienes que ver estos parques donde mas de una montana...	['disfrutan', 'emociones', 'fuertes', 'entonces', 'ver', 'parques', u'ma', 'montana']
3	POST:	!buenas noches, viajeros! que suenen con ciudades fantasticas y aventuras inolvidables.	['viajeros', 'suenen', 'ciudades', 'fantasticas', 'aventuras', 'inolvidables']
4	COMMENT:	@travelreportmx la hermosisima ciudad luz, paris.!!!	['hermosisima', 'ciudad', 'luz', 'paris']
5	POST:	manita arriba si iniciaste el dia asi por lo que te trajeron los #reyesmagos	['si', 'iniciaste', 'dia', 'asi', 'trajeron', 'reyesmagos']
6	POST:	tu periodico semanal !travel report esta disponible! #guanajuato #buenprovecho	['semanal', 'travel', 'report', 'disponible', 'guanajuato', 'buenprovecho']
7	POST:	hoy #6deenero comienza el tour ideal para los amantes de los #helados. !recorre las mejores heladerias de la...	['comienza', 'tour', 'ideal', 'amantes', 'helados', 'recorre', 'mejores', 'heladerias']
8	POST:	?que tal una pequena encuesta? ?donde te gustaba mas tomarte las fotografias con los #reyesmagos ?	['pequena', 'encuesta', 'gustaba', u'ma', 'tomarte', 'fotografias', 'reyesmagos']
9	POST:	?quien ira al #festival de la rosca y el #chocolate? #parquedelosvenados !a comer!	['festival', 'rosca', 'chocolate', 'parquedelosvenados', 'comer']
10	POST:	?estan listos para leer todo el libro? o solamente se quedaran con una pagina...	['leer', 'libro', 'solamente', 'quedaran', 'pagina']
11	POST:	si en tu viaje tienes un companero que no para de sacarse selfies es porque... #millennials	['viaje', 'companero', 'sacarse', 'selfies', 'millennials']
12	POST:	este es el #audiorama de #chapultepec, perfecto para escuchar musica, leer o meditar sin vendedores ambulantes...	['audiorama', 'chapultepec', 'perfecto', 'escuchar', 'musica', 'leer', 'meditar', 'vendedores', 'ambulantes']
13	POST:	?escalera o cerca de un gato negro? tambien los viajeros tienen sus #supersticiones	['cerca', 'gato', 'negro', 'tambien', 'viajeros', 'supersticiones']
14	POST:	!inicia tu #domingo con un pequeno recordatorio!	['domingo', 'pequeno', 'recordatorio']

Fig. 4-5 Ejemplo de normalización de tweets de la cuenta TravelReportMX [Elaboración propia]

En la siguiente Fig. 4-6, se muestra el flujo que tiene el objeto MongoDB_ExtractInfo que se utiliza para extraer los *tweets* guardados en MongoDB. Para tener la información organizada, se crea un archivo XLS en el cual se guardará cada *post&comments* (normalizados) en forma de *timeline*. Para esto, se eliminan todos los *retweets* ya que es información duplicada, se realiza una búsqueda de emoticones, se realiza una búsqueda por ID de tweets para relacionar que *comments* pertenecen a cada *post* y si se encuentra alguno se reemplaza por texto, se normaliza la información (tweets) de manera que se elimina y modifica texto que no altera la importancia de los mensajes (como: convertir texto a letras minúsculas, eliminar signos de puntuación, eliminar URL, convertir números a texto) y finalmente, se guarda cada tweet en el archivo XLS.

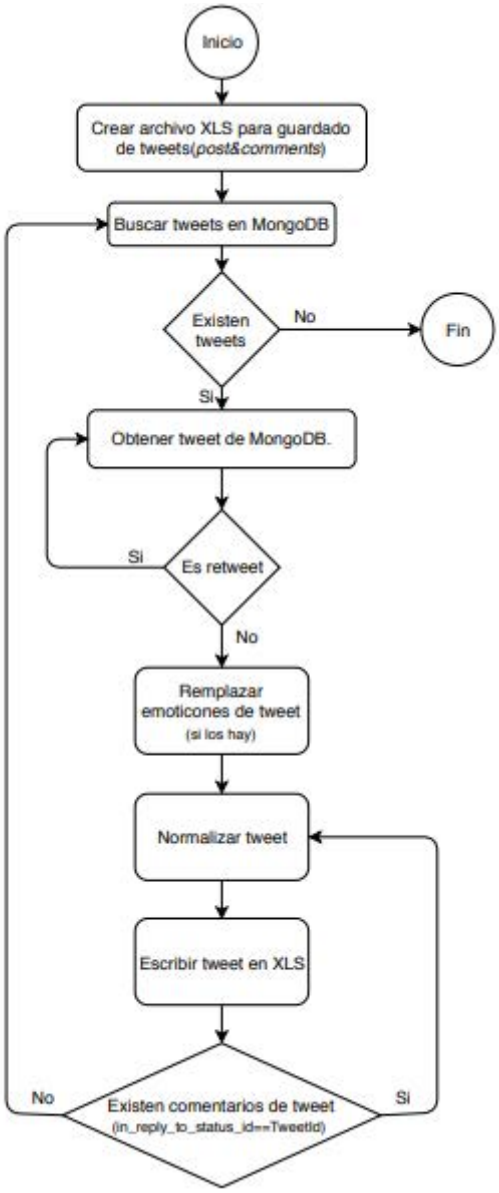


Fig. 4-6 Objeto MongoDB_ExtractInfo. [Elaboración propia]

4.3 Clasificación de tweets con algoritmos SVM

SVM (Suppport Vector Machine) es uno de los algoritmos que han dado mejores resultados en clasificación de sentimientos. Éste proceso se lleva a cabo con aprendizaje supervisado y su principal característica es que maneja el texto como vectores de soporte. Para éste proyecto, SVM (véase Figura 3.8) ve a cada *tweet* como un vector en un espacio vectorial, donde por medio de una "*bag of words*" se genera una matriz de vectores, los cuales son separados por medio del aprendizaje supervisado (nada

recomendado, no recomendado, neutro, recomendado, muy recomendado); el algoritmo SVM es entrenado con dichos datos, para posteriormente probar con otros datos, como se muestra en la Fig. 4-7.

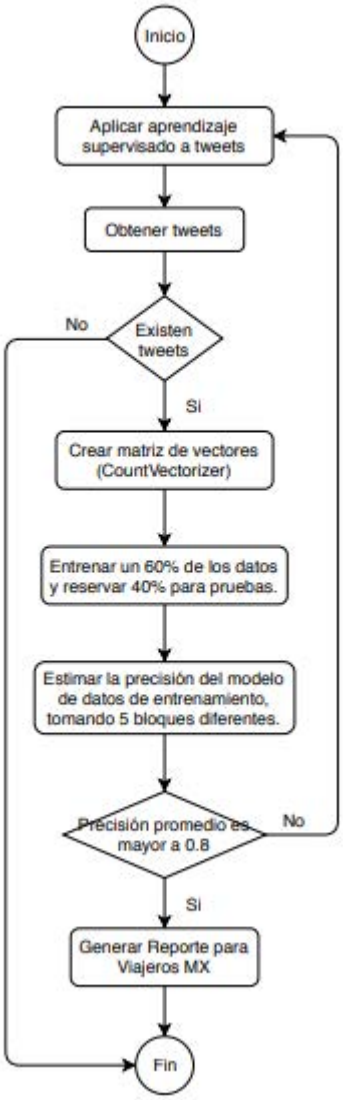


Fig. 4-7 SVM object. [Elaboración propia]

4.4 Generación Web de Reporte de Viajeros MX

Utilizando el aprendizaje supervisado y la clasificación del algoritmo SVM se genera un reporte Web de ayuda a viajeros, para el cual se utiliza una herramienta llamada DBpedia, la cual es un servicio de base de datos que se conecta a un servidor remoto (de Wikipedia) que por su naturaleza ayuda a determinar la localidad de los *tweets* tomando en cuenta el contenido de éste, ya sea que esté mencionando algún estado de México, otro lugar o lugares como restaurantes, que la herramienta no ubique su localidad.

La instalación y configuración del servicio BDpedia corre bajo un ambiente java en un servidor Apache Tomcat y necesita de un sistema operativo Linux. El ambiente debe contar con memoria RAM suficiente para levantar el servicio en el servidor.

Para levantar localmente el servicio DBpedia en el servidor Tomcat, se debe ejecutar el comando:

java -jar dbpedia-spotlight-1.0.0.jar path/to/model/folder/en_2+2 http://localhost:2222/rest

como se muestra en la Fig. 4-8:

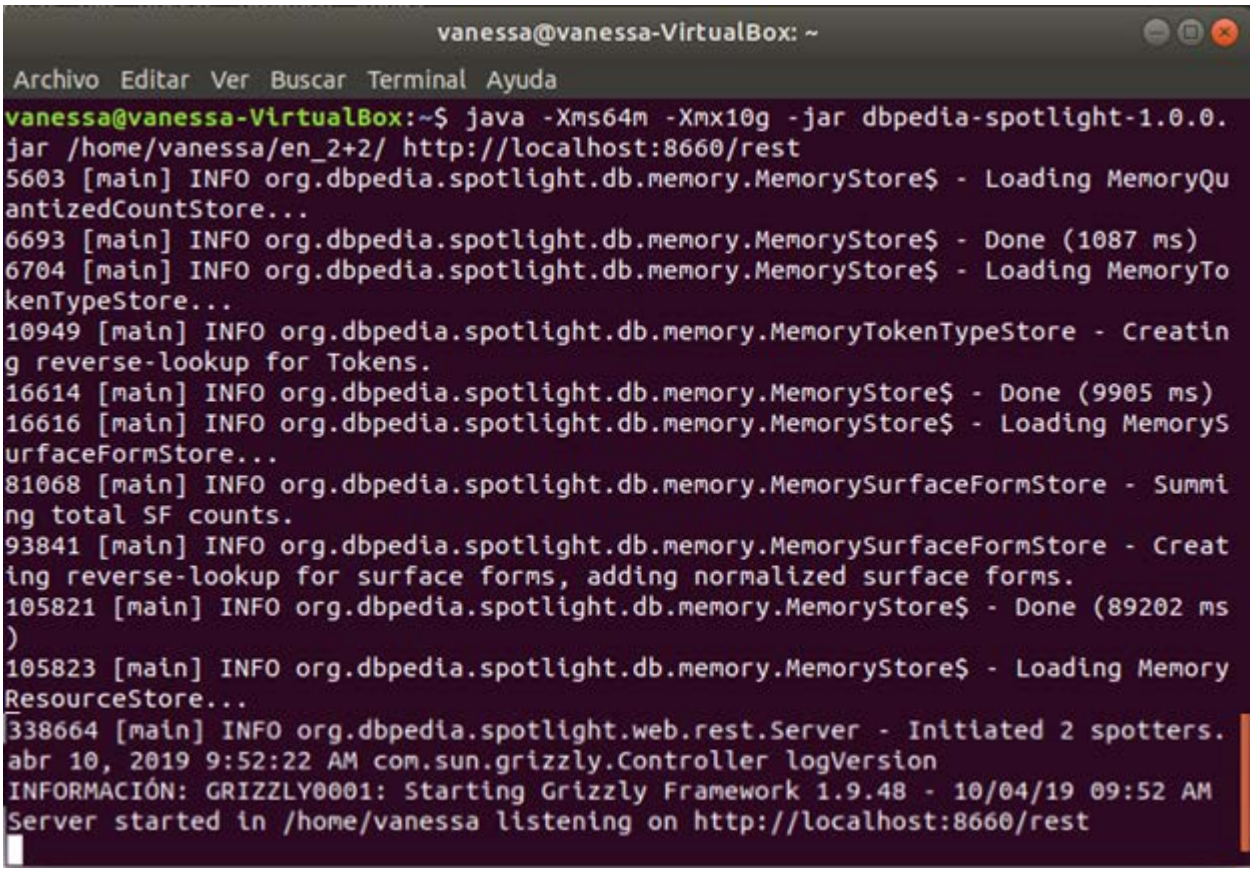


Fig. 4-8 Servidor DBpedia. [Elaboración propia]

Para saber que el servidor DBpedia está funcionando correctamente, se realiza un test con el comando *curl* y así mismo se puede observar la respuesta de la base de datos DBpedia en estructura XML (véase Fig. 4-9).

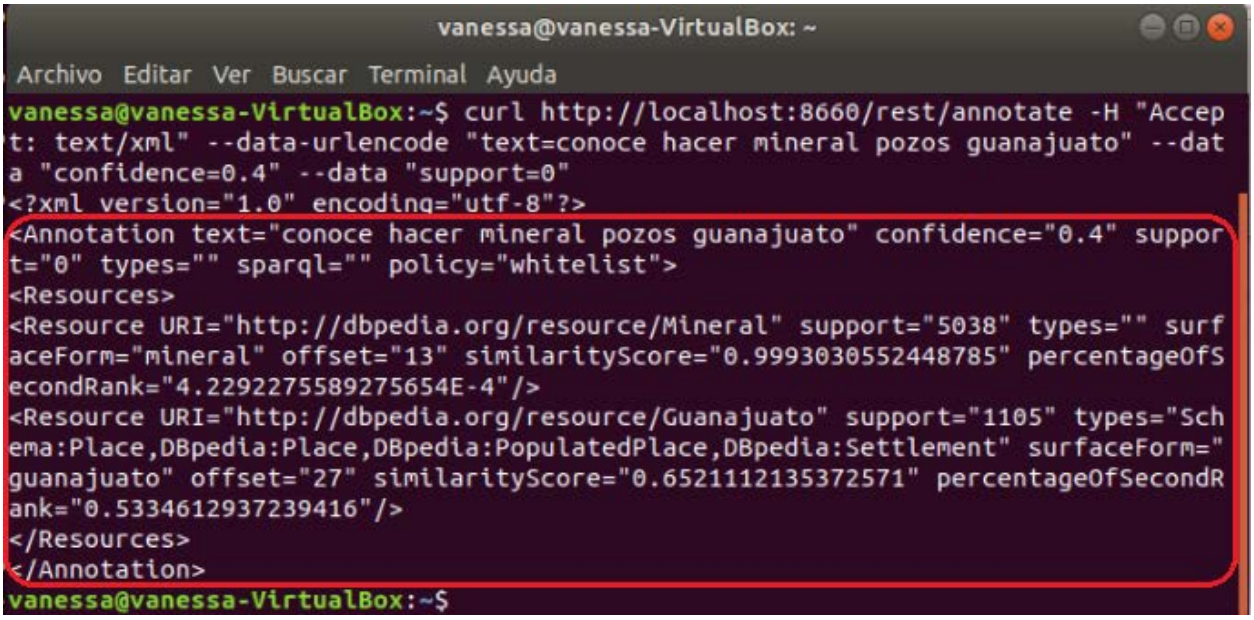


Fig. 4-9 Respuesta XML de base de datos BDPedia [Elaboración propia]

El flujo del reporte web de Viajeros MX, se desarrolló para entender como es el flujo del algoritmo desarrollado para mostrar a los usuarios viajeros (usuarios finales) los *tweets* por estados mexicanos u otros lugares (pudiendo ser, otros países, nombres de restaurantes, etc.) y por recomendación (dado por el AS y validado por SVM), y finalmente saber cómo es que Flask ayuda a mostrar la información en la web (véase Fig. 4-10).

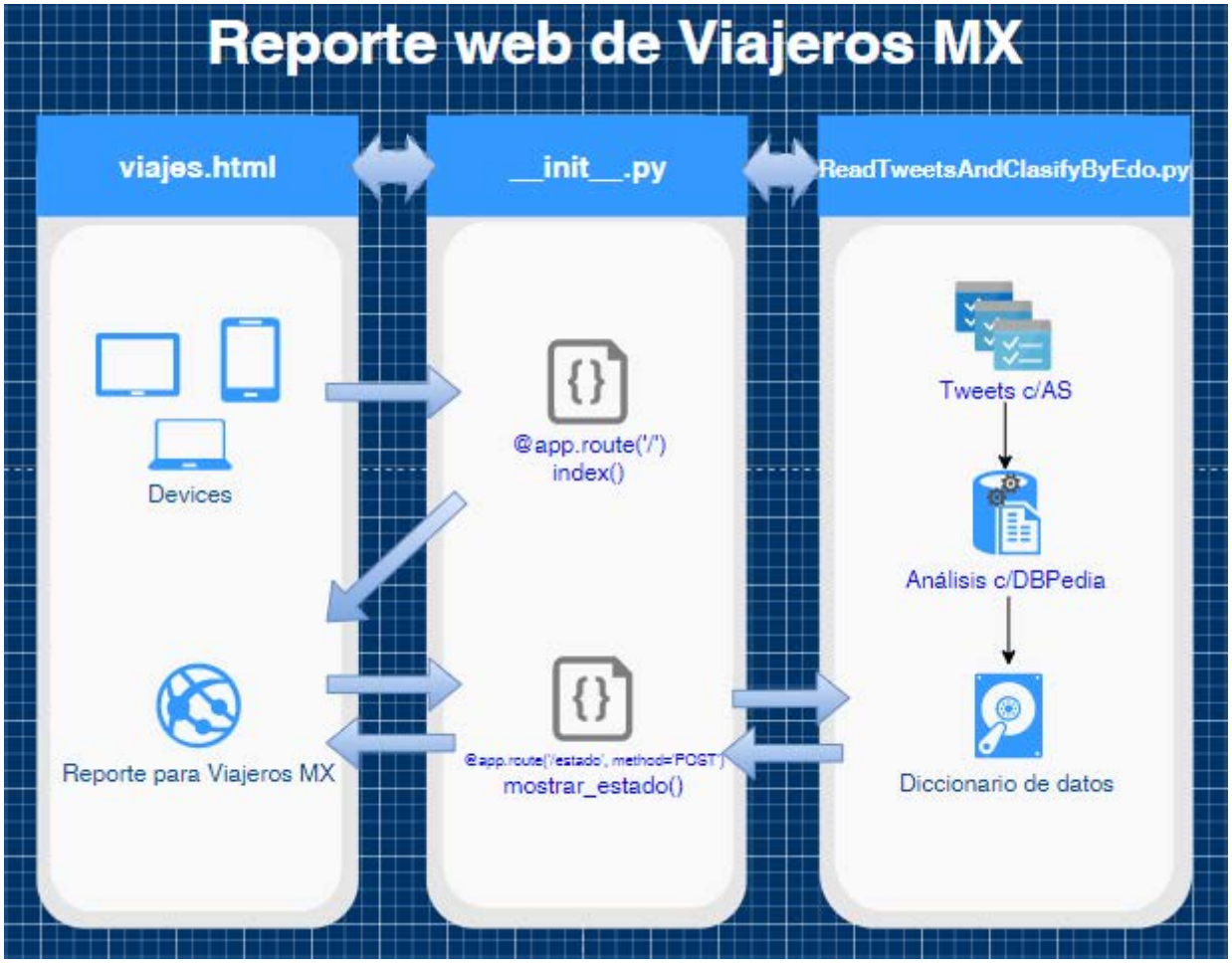


Fig. 4-10 Flujo de reporte web de Viajeros MX [Elaboración propia]

5) CAPITULO V. PRUEBAS Y RESULTADOS

La generación del reporte “Reporte de viajeros MX” muestra como resultado, un análisis en el cual se puede observar que la información obtenida de los *tweets* de usuarios viajeros por sitios mexicanos, es de utilidad para otros viajeros, que destinos son los más concurridos y saber que tan buena es o puede ser la calidad de la información que aportan los usuarios de Twitter para conocer o no algún destino turístico.

Para la obtención de éstos resultados, se eliminó el ruido que genera la obtención de *tweets* de la base de datos MongoDB, se normalizaron los *tweets*, se aplicó Aprendizaje Supervisado a cada tweet para poder entrenar y validar el algoritmo SVM que ayuda a definir las categorías (Nada recomendado, No tan recomendado, Recomendado y No recomendados) pertenecientes cada tweet y da un *score* diciendo que tan buena o mala fue la categorización del Aprendizaje Supervisado.

En la normalización de tweets, se tomó en cuenta: eliminación de ruido (corchetes, brakets, tags de objeto JSON), convertir todo el texto en minúsculas, cambiar números por texto, eliminar URL, eliminar puntuación (‘.’,’:’,’…’,’,’,’;’, ‘@’,’#’ etc.), eliminar *stopwords* y se aplicó lematización de palabras. Esto con la finalidad de determinar qué tan bueno o malo fue el AS y la normalización que se aplicaron; se realizaron varias pruebas del algoritmo SVM usando el procedimiento K-fold (*CrossValidation*) para probar los datos y ayudar a determinar el mejor camino para presentar la información en el reporte final. Las pruebas que fueron aplicadas en los tweets (corpus), tomaron en cuenta los siguientes factores para un AS correcto y erróneo:

- Tweets sin normalizar
- Tweets normalizados con lematización
- Tweets normalizados sin lematización
- Usando bigramas
- Usando trigramas.

Con un corpus total de 3899 tweets, finalmente se muestra a continuación el análisis y resultados que se realizó para las 3 cuentas de Twitter (TravelReportMX, MexDesconocido y WorldThruMyEyes):

5.1 TRAVEL REPORT MX

5.1.1 Corpus con Aprendizaje Supervisado correcto

La Fig. 5-1 proporciona gráficamente los resultados del Aprendizaje Supervisado (por cantidad de tweets y categoría) en el corpus obtenido:

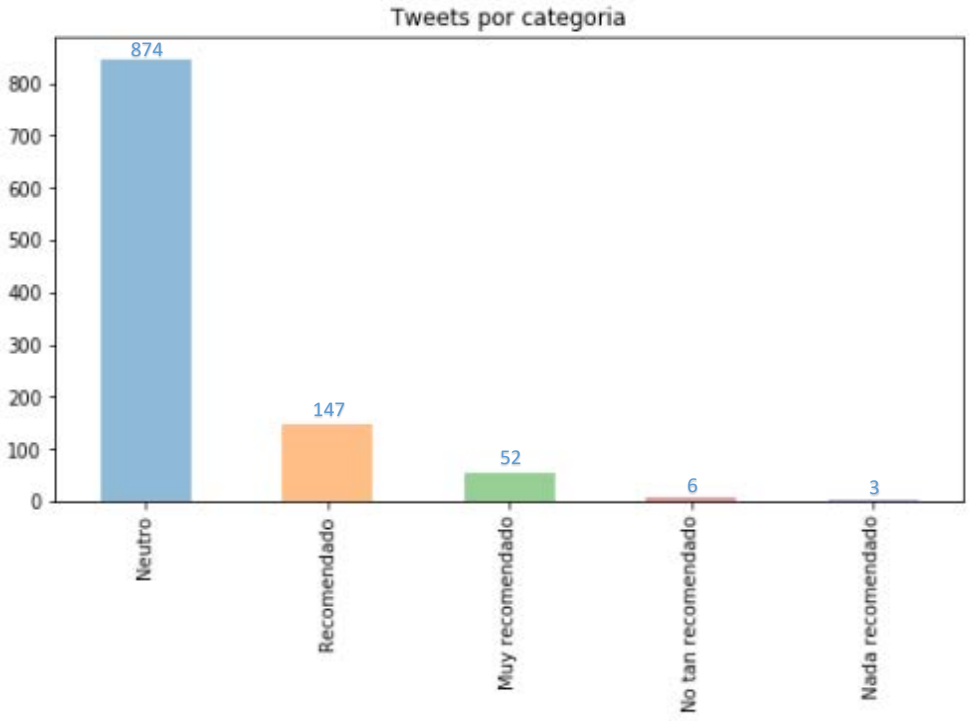


Fig. 5-1 Cantidad de Tweets por categoría TravelReportMX correcto [Elaboración propia]

En la Fig. 5-2 se observa un ejemplo de cómo se visualiza la información aplicando cada uno de los factores que se evaluaron para el procesamiento del algoritmo SVM con un Aprendizaje Supervisado correcto:

POSTING	tweets	normalized_tweet	bigrams	trigrams	AS
POST:	levante la mano quien quiere ir a #teotihuacan. !a seguir estos #mandamientos! ?ya has ido?	['levante', 'mano', 'quiere', 'ir', 'teotihuacan', 'seguir', 'mandamientos', 'ido']	['levante mano', 'mano quiere', 'quiere ir', 'ir teotihuacan', 'teotihuacan seguir', 'seguir mandamientos', 'mandamientos ido']	['levante mano quiere', 'mano quiere ir', 'quiere ir teotihuacan', 'ir teotihuacan seguir', 'teotihuacan seguir mandamientos', 'seguir mandamientos ido']	1
COMMENT:	@travelreportmx @sanxpm el tour en bici esta padrisimo! recomendamos con nuestros amigos de @teoenbici	['travelreportmx', 'sanxpm', 'tour', 'bici', 'padrisimo', 'recomendamos', 'amigos', 'teoenbici']	['travelreportmx sanxpm', 'sanxpm tour', 'tour bici', 'bici padrisimo', 'padrisimo recomendamos', 'recomendamos amigos', 'amigos teoenbici']	['travelreportmx sanxpm tour', 'sanxpm tour bici', 'tour bici padrisimo', 'bici padrisimo recomendamos', 'padrisimo recomendamos amigos', 'recomendamos amigos teoenbici']	2
COMMENT:	@travelreportmx creo que estas siguiendo mis pasos, fui ayer con mis amigos y sobrino!	['travelreportmx', 'creo', 'siguiendo', 'pasos', 'ayer', 'amigos', 'sobrino']	['travelreportmx creo', 'creo siguiendo', 'siguiendo pasos', 'pasos ayer', 'ayer amigos', 'amigos sobrino']	['travelreportmx creo siguiendo', 'creo siguiendo pasos', 'siguiendo pasos ayer', 'pasos ayer amigos', 'ayer amigos sobrino']	0
COMMENT:	@travelreportmx yo!yo!yo!yo!	['travelreportmx', 'yoyoyoyo']	['travelreportmx yoyoyoyo']	[]	0

Fig. 5-2 Ejemplo de información TravelReportMX procesada con AS correcto. [Elaboración propia]

Aplicando al algoritmo SVM los factores mencionados anteriormente, los resultados K-fold son los mostrados en la Tabla 4.

Tabla 4 Resultados TravelReportMX con AS correcto [Elaboración propia]

TRAVEL REPORT MX c/AS correcto					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.798195905597996	0.8038391544095319	0.8038391544095319	0.8010130577319842	0.7972437753711527
	0.80 (+/- 0.03)	0.80 (+/- 0.02)	0.80 (+/- 0.02)	0.80 (+/- 0.02)	0.80 (+/- 0.03)

En la Fig. 5-3 se muestra un ejemplo real de los datos que muestra el programa ejecutado en la consola, que es lo que se observa sombreado en la columna 2 de la Tabla 4. Datos que por ser los de mejor precisión, serán tomados para la generación del reporte para viajeros.

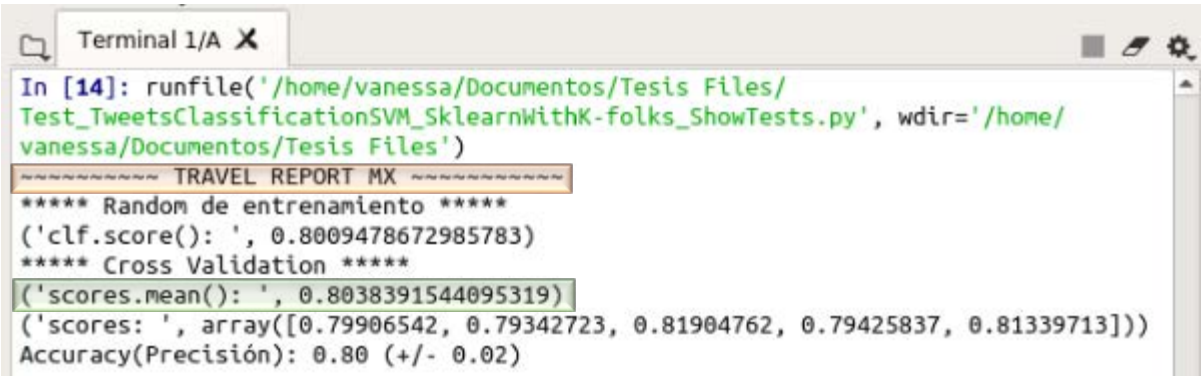


Fig. 5-3 Scores del entrenamiento de TravelReportMX desde consola. [Elaboración propia]

5.1.2 Corpus con Aprendizaje Supervisado erróneo

La Fig. 5-4 proporciona gráficamente los resultados del Aprendizaje Supervisado erróneo (por cantidad de tweets y categoría) en el corpus obtenido, donde se puede observar ligeramente el cambio de cantidad de tweets por categoría.

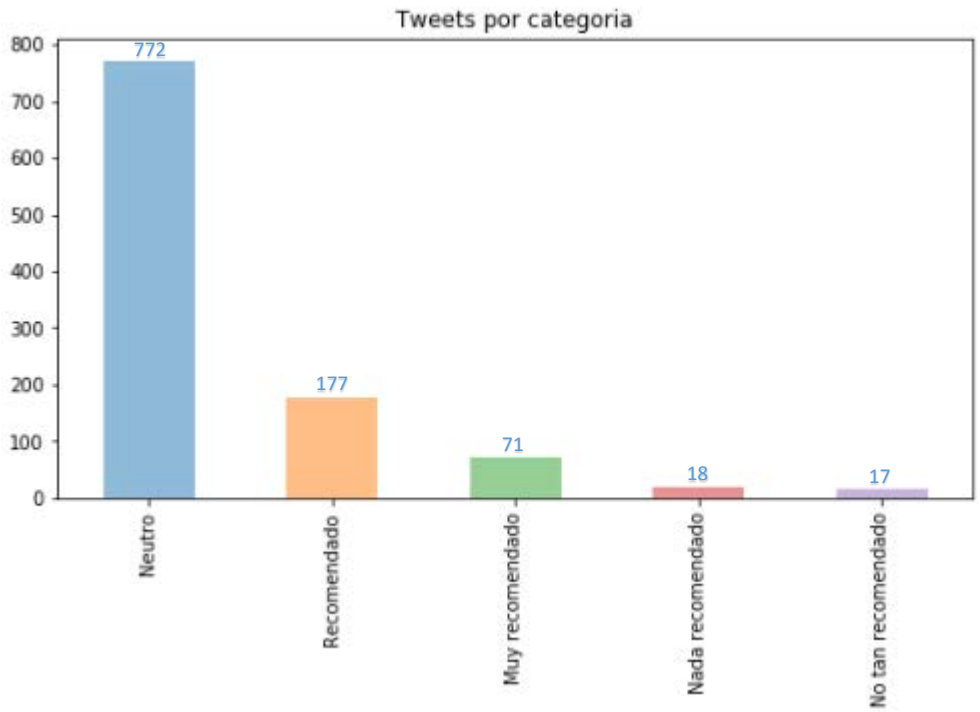


Fig. 5-4 Cantidad de Tweets por categoría TravelReportMX erróneo [Elaboración propia]

En la Fig. 5-5 se proporciona un ejemplo de cómo se visualiza la información aplicando cada uno de los factores que se evaluaron para el procesamiento del algoritmo SVM con un Aprendizaje Supervisado erróneo:

POSTING	tweets	normalized_tweet	bigrams	trigrams	AS
POST:	levante la mano quien quiere ir a #teotihuacan. !a seguir estos #mandamientos! ?ya has ido?	['levante', 'mano', 'quiere', 'ir', 'teotihuacan', 'seguir', 'mandamientos', 'ido']	['levante mano', 'mano quiere', 'quiere ir', 'ir teotihuacan', 'teotihuacan seguir', 'seguir mandamientos', 'mandamientos ido']	['levante mano quiere', 'mano quiere ir', 'quiere ir teotihuacan', 'ir teotihuacan seguir', 'teotihuacan seguir mandamientos', 'seguir mandamientos ido']	1
COMMENT:	@travelreportmx @sanxpm el tour en bici esta padrisimo! recomendamos con nuestros amigos de @teoenbici	['travelreportmx', 'sanxpm', 'tour', 'bici', 'padrisimo', 'recomendamos', 'amigos', 'teoenbici']	['travelreportmx sanxpm', 'sanxpm tour', 'tour bici', 'bici padrisimo', 'padrisimo recomendamos', 'recomendamos amigos', 'amigos teoenbici']	['travelreportmx sanxpm tour', 'sanxpm tour bici', 'tour bici padrisimo', 'bici padrisimo recomendamos', 'padrisimo recomendamos amigos', 'recomendamos amigos teoenbici']	1
COMMENT:	@travelreportmx creo que estas siguiendo mis pasos, fui ayer con mis amigos y sobrino!	['travelreportmx', 'creo', 'siguiendo', 'pasos', 'ayer', 'amigos', 'sobrino']	['travelreportmx creo', 'creo siguiendo', 'siguiendo pasos', 'pasos ayer', 'ayer amigos', 'amigos sobrino']	['travelreportmx creo siguiendo', 'creo siguiendo pasos', 'siguiendo pasos ayer', 'pasos ayer amigos', 'ayer amigos sobrino']	1
COMMENT:	@travelreportmx yo!yo!yo!yo!	['travelreportmx', 'yoyoyoyo']	['travelreportmx yoyoyoyo']	[]	1

Fig. 5-5 Ejemplo de información TravelReportMX procesada con AS erróneo [Elaboración propia]

Aplicando al algoritmo SVM los factores mencionados anteriormente, los resultados K-fold se observan en la Tabla 5.

Tabla 5 Resultados TravelReportMX con Aprendizaje Supervisado erróneo [Elaboración propia]

TRAVEL REPORT MX c/AS erróneo					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.7044014100252366	0.7148602308600369	0.7148602308600369	0.6948766671786146	0.711962671406219
	0.70 (+/- 0.04)	0.71 (+/- 0.05)	0.71 (+/- 0.05)	0.69 (+/- 0.05)	0.71 (+/- 0.04)

Realizando un comparativo entre los resultados de las *Tabla 4* y *5,* se puede observar que un Aprendizaje Supervisado bien hecho es importante para el mejor funcionamiento del algoritmo SVM en la clasificación de información. Así mismo, tomando en cuenta los resultados ***K-fold*** del **corpus con Aprendizaje Supervisado correcto** como se observa en la *Tabla 5*, es posible visualizar que el procesamiento en los "Tweets normalizados s/lematización" y los "Tweets normalizados c/lematización" son las pruebas que muestran una mejor precisión y menor margen de error.

5.2 MEXDESCONOCIDO:

5.2.1 Corpus con Aprendizaje Supervisado correcto

La Fig. 5-6 muestra gráficamente los resultados del Aprendizaje Supervisado (por cantidad de tweets y categoría) en el corpus obtenido.

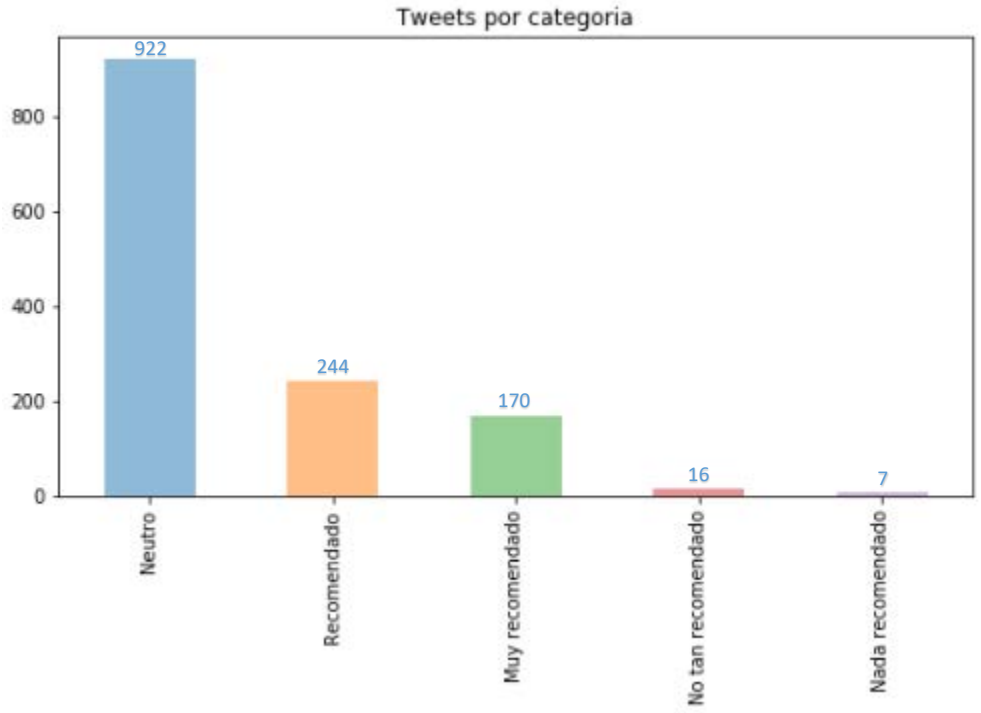


Fig. 5-6 Cantidad de Tweets por categoría MexDesconocido con AS correcto [Elaboración propia]

En la Fig. 5-6 se observa un ejemplo de cómo se visualiza la información aplicando cada uno de los factores que se evaluaron para el procesamiento del algoritmo SVM con un Aprendizaje Supervisado correcto.

POSTING	tweets	normalized_tweet	bigrams	trigrams	AS
POST:	3 playas de mexico para viajar en temporada baja -	['tres', 'playas', 'mexico', 'viajar', 'temporada', 'baja']	['tres playas', 'playas mexico', 'mexico viajar', 'viajar temporada', 'temporada baja']	['tres playas mexico', 'playas mexico viajar', 'mexico viajar temporada', 'viajar temporada baja']	1
COMMENT:	@mexdesconocido en temporada baja, cualquier playa en mexico es una excelente idea!	['mexdesconocido', 'temporada', 'baja', 'cualquier', 'playa', 'mexico', 'excelente', 'idea']	['mexdesconocido temporada', 'temporada baja', 'baja cualquier', 'cualquier playa', 'playa mexico', 'mexico excelente', 'excelente idea']	['mexdesconocido temporada baja', 'temporada baja cualquier', 'baja cualquier playa', 'cualquier playa mexico', 'playa mexico excelente', 'mexico excelente idea']	2
COMMENT:	@mexdesconocido me late amor! en bacalar una amiga tiene un hotel, y en tulum creo que tambien, de hecho. me encanta la idea mi amor!	['mexdesconocido', 'late', 'amor', 'bacalar', 'amiga', 'hotel', 'tulum', 'creo', 'tambien', 'hecho', 'encanta', 'idea', 'amor']	['mexdesconocido late', 'late amor', 'amor bacalar', 'bacalar amiga', 'amiga hotel', 'hotel tulum', 'tulum creo', 'creo tambien', 'tambien hecho', 'hecho encanta', 'encanta idea', 'idea amor']	['mexdesconocido late amor', 'late amor bacalar', 'amor bacalar amiga', 'bacalar amiga hotel', 'amiga hotel tulum', 'hotel tulum creo', 'tulum creo tambien', 'creo tambien hecho', 'tambien hecho encanta', 'hecho encanta idea', 'encanta idea amor']	2
COMMENT:	@mexdesconocido favor de enviar mas informacion. saludos	['mexdesconocido', 'favor', 'enviar', 'mas', 'informacion', 'saludos']	['mexdesconocido favor', 'favor enviar', 'enviar mas', 'mas informacion', 'informacion saludos']	['mexdesconocido favor enviar', 'favor enviar mas', 'enviar mas informacion', 'mas informacion saludos']	0

Fig. 5-7 Ejemplo de información MexDesconocido procesada con Aprendizaje Supervisado correcto [Elaboración propia]

Aplicando al algoritmo SVM los factores mencionados anteriormente, los resultados *K-fold* se muestran en la Tabla 6.

Tabla 6 Resultados MexDesconocido con Aprendizaje Supervisado correcto [Elaboración propia]

MEX DESCONOCIDO c/AS correcto					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.7099828955028674	0.6952819816051427	0.6952738415970027	0.6732008768284256	0.6974581572130527
	0.71 (+/- 0.04)	0.70 (+/- 0.04)	0.70 (+/- 0.04)	0.67 (+/- 0.05)	0.70 (+/- 0.05)

5.2.2 *Corpus con Aprendizaje Supervisado mejorado*

Debido a que los resultados del apartado 5.2.1 no fueron tan exitosos, se volvió a realizar Aprendizaje Supervisado para ésta cuenta y efectivamente, las puntuaciones mejoraron (por cambios que hubo en las categorías Neutro, Recomendado y Muy recomendado), como puede verse en la Fig. 5-8.

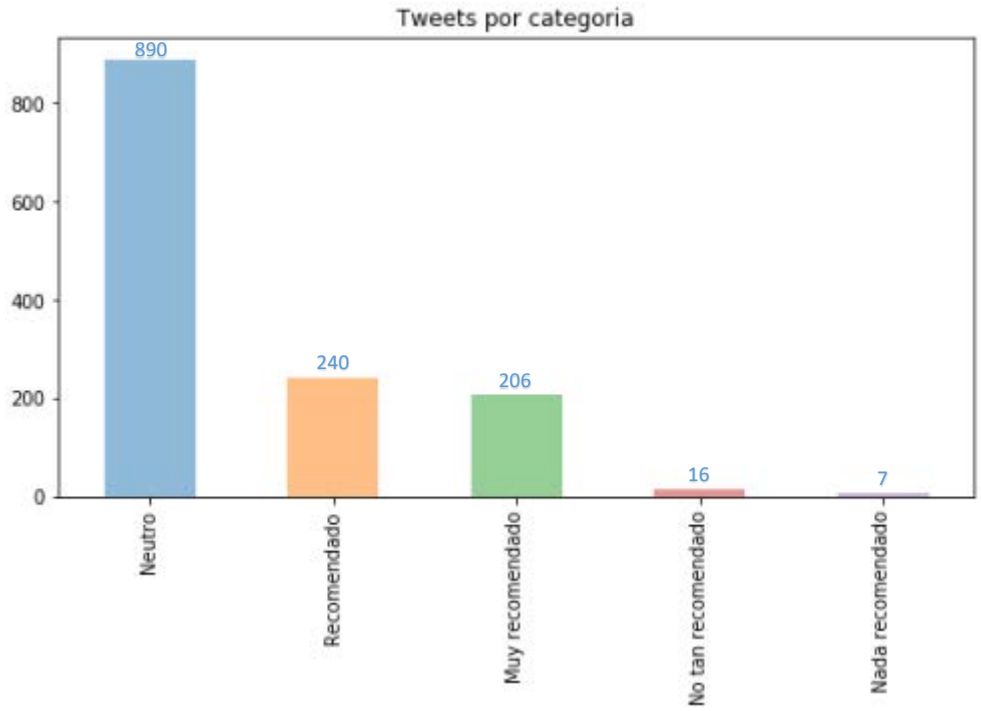


Fig. 5-8 Cantidad de Tweets por categoría MexDesconocido con Aprendizaje Supervisado mejorado [Elaboración propia]

Un ejemplo de los cambios que se realizaron en el Aprendizaje Supervisado se observa en la Fig. 5-9:

AS inicial:

POST:	#findesemana > no te pierdas el festival nacional de vuelo de papalotes	0
COMMENT:	@mexdesconocido ojala siguiera esa tradicion, porque ya se perdio	0
COMMENT:	@mexdesconocido buenisimo	0

AS mejorado:

POST:	#findesemana > no te pierdas el festival nacional de vuelo de papalotes	1
COMMENT:	@mexdesconocido ojala siguiera esa tradicion, porque ya se perdio	1
COMMENT:	@mexdesconocido buenisimo	2

Fig. 5-9 Mejoramiento del Aprendizaje Supervisado en MexDesconocido [Elaboración propia]

Mejorando la precisión (resultados de las pruebas K-fold) los resultados se muestran en la Tabla 7.

Tabla 7 Resultados MexDesconocido con Aprendizaje Supervisado mejorado. [Elaboración propia]

MEX DESCONOCIDO c/AS mejorado					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.730491266859856	0.7202615545369709	0.7202615545369709	0.6988619943627516	0.6973806124922564
	0.73 (+/- 0.10)	0.72 (+/- 0.07)	0.72 (+/- 0.07)	0.70 (+/- 0.11)	0.70 (+/- 0.11)

En la Fig. 5-10 se muestra un ejemplo real de los datos que muestra el programa ejecutado en la consola, que es lo que se observa en la columna 2 de la Tabla 7. Datos que no son los de mayor precisión, pero son datos normalizados que no varían mucho de los datos con mayor precisión y serán tomados en cuenta para la generación del reporte para viajeros.

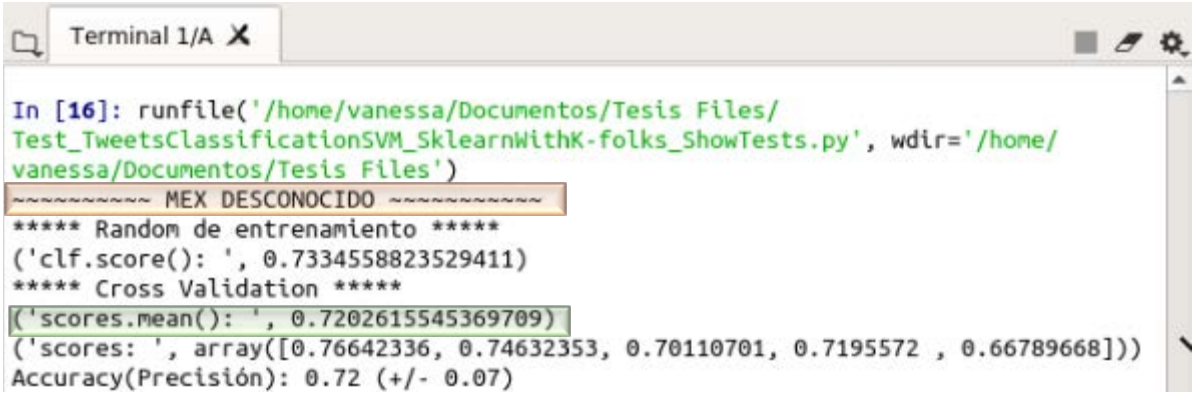


Fig 5-10 Scores del entrenamiento de MexDesconocido desde consola. [Elaboración propia]

Como se puede observar en la Tabla 7, los resultados del *Aprendizaje Supervisado correcto* mejoraron (*Aprendizaje Supervisado mejorado)*, pero no lo suficiente para obtener la precisión deseada. Realizando de nuevo un análisis de la información, se observó que existen varios tweets que se tomaron como neutros, aunque parecieran buenas recomendaciones ya que la información del tweet se apoya de imágenes o videos. A continuación, se muestran algunos ejemplos:

- te encantara conocer esta pequena isla en nayarit.
- te dará 4 rutas por veracruz, para que disfrutes de lo mejor de este increible estado.
- un lugar pequeno en donde viviras grandes experiencias: el bar mas pequeno de san miguel de allende...
- rodeado de espeso bosque y bellos paisajes se encuentra este centro ceremonial que te sorprendera....
- si lo tuyo es la adrenalina y la aventura, en estos pueblos magicos encontraras la dosis exacta que necesitas....
- planea una visita a alguno -o a todos- de estos laberintos. la experiencia te encantara.

5.2.3 Corpus con Aprendizaje Supervisado erróneo

Otra forma de probar que el Aprendizaje Supervisado es importante en la clasificación de información de algoritmo SVM, es mostrando la misma información sin análisis de información, con un Aprendizaje Supervisado erróneo (distinto) (véanse Fig. 5-11 y Tabla 8).

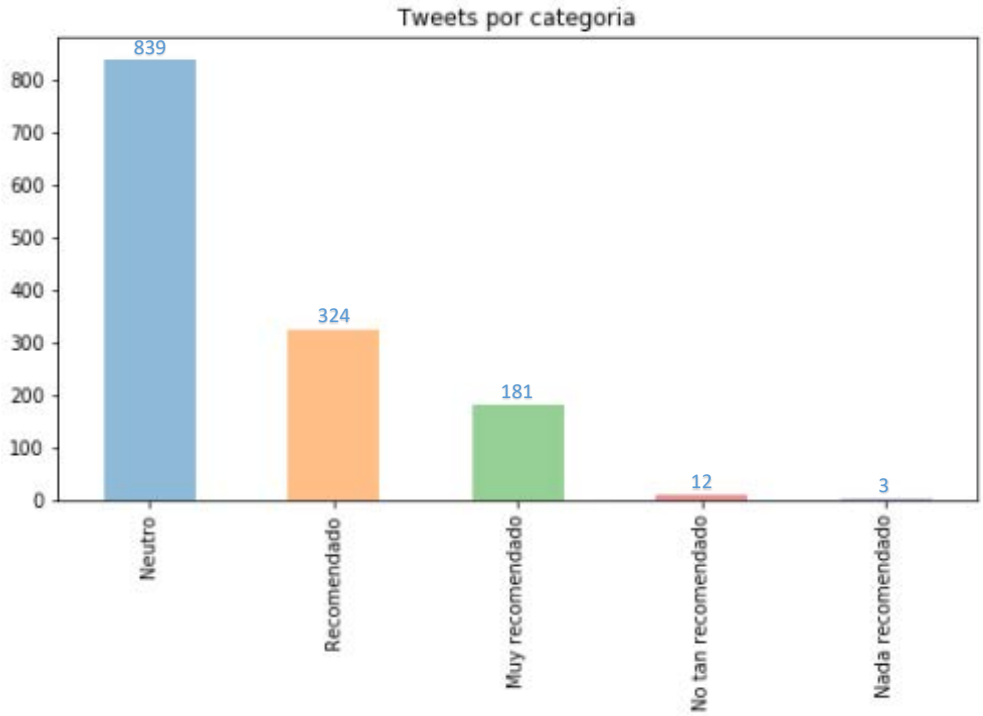


Fig. 5-11 Cantidad de Tweets por categoría MexDesconocido con Aprendizaje Supervisado erróneo [Elaboración propia]

En la Tabla 8 se observa un ejemplo de Aprendizaje Supervisado erróneo.

Tabla 8 Ejemplo de MexDesconocido con Aprendizaje Supervisado erróneo [Elaboración propia]

A	B	E
POSTING	TWEETS	AS
POST:	?tienes poco tiempo para una escapada? el pueblo magico de tequila es un destino perfecto para un viaje expres....	0
POST:	?tienes pocos dias para viajar? te recomendamos visitar la ciudad de campeche. te damos una guia para que recorras...	0
POST:	el impresionante vitromural de zacatlan de las manzanas -	1
POST:	el impresionante vitromural de zacatlan de las manzanas -	2

Donde los resultados de precisión (K-fold) del algoritmo SVM se encuentran visualizados en la Tabla 9.

Tabla 9 Resultados MexDesconocido con Aprendizaje Supervisado erróneo. [Elaboración propia]

MEX DESCONOCIDO c/AS erróneo					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.6429512766594284	0.6341626033224003	0.6319404402905399	0.6121382376348482	0.630480666672186
	0.64 (+/- 0.06)	0.63 (+/- 0.06)	0.63 (+/- 0.06)	0.61 (+/- 0.04)	0.63 (+/- 0.06)

A diferencia de TRAVELREPORT, MEXDESCONOCIDO da un mejor score usando Tweets s/normalizar en los 3 casos presentados por una pequeña diferencia de cuando se aplica normalización.

5.3 WORLDTHRUMYEYES

5.3.1 *Corpus con Aprendizaje Supervisado correcto*

La Fig. 5-12 muestra gráficamente los resultados del Aprendizaje Supervisado (por cantidad de tweets y categoría) del corpus obtenido:

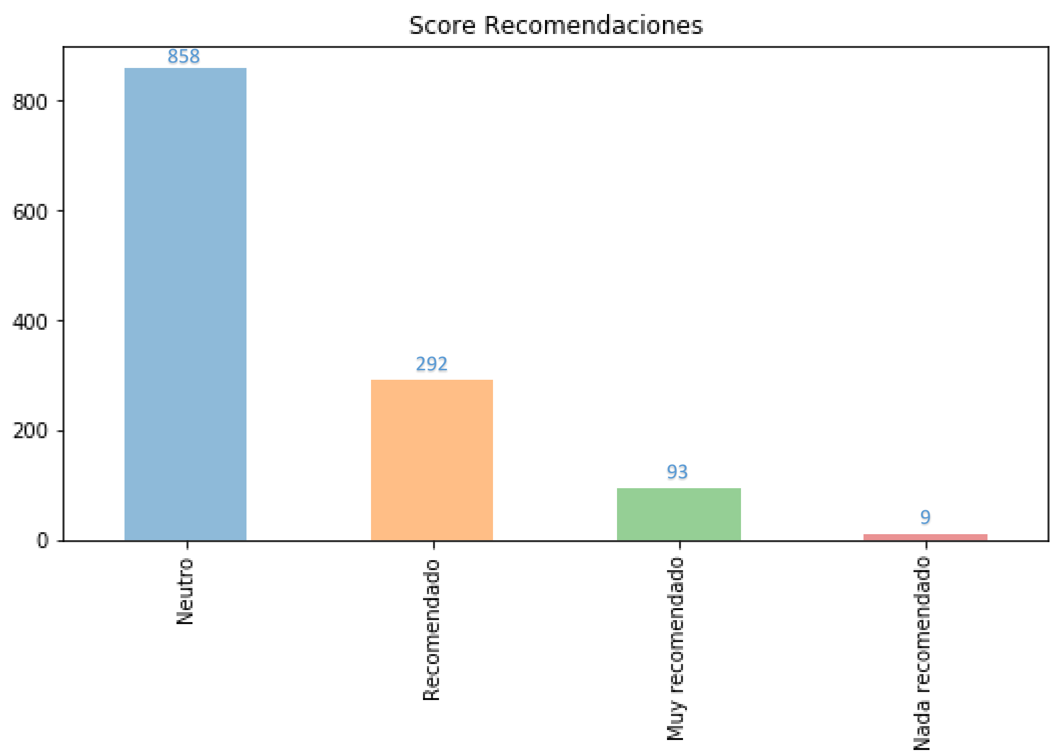


Fig. 5-12 Cantidad de Tweets por categoría WorldThruMyEyes con Aprendizaje Supervisado correcto [Elaboración propia]

A diferencia de las otras 2 cuentas de Twitter, la cuenta *WorldThruMyEyes* mostró muchas repeticiones en sus tweets, (Véanse Fig. 5-13 y 5-14).

POSTING	TWEETS	AS
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0
POST:	? por que necesito un seguro de viajes ? creeme lo necesitas y te cuento por que	0

Fig. 5-13 Tweets WorldThruMyEyes repetidos [Elaboración propia]

POSTING	TWEETS	AS
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2
POST:	esta maravilla son las tumbas licias de myra en turquia: necropolis rupestre	2

Fig. 5-14 Tweets WorldThruMyEyes repetidos 2 [Elaboración propia]

Esta información recuperada generó que WorldThruMyEyes (tweets repetidos) diera un score "muy alto" y a simple vista "muy bueno", debido a que el aprendizaje supervisado que se realizó en algunos casos por lo menos con triplicidad de datos; esto significa que el algoritmo SVM detecta que es exacto el Aprendizaje Supervisado y por lo cual, da un mejor resultado. Véanse en la Tabla 10 los resultados de precisión *K-fold*.

Tabla 10 Resultados CrossVal-WorldThruMyEyes con Aprendizaje Supervisado correcto [Elaboración propia]

WorldThruMyEyes c/AS correcto					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.9912123146965743	0.9920059654902251	0.9920059654902251	0.9912123146965743	0.9904186639029235
	0.99 (+/- 0.01)	0.99 (+/- 0.01)	0.99 (+/- 0.01)	0.99 (+/- 0.01)	0.99 (+/- 0.02)

5.3.2 *WorldThruMyEyes resumido*

Las pruebas mostradas en este apartado se realizaron eliminando las repeticiones que se observaron en el apartado 5.3.1, pretendiendo obtener resultados "más" reales del algoritmo SVM; en la Fig. 5-15 los cambios se ven reflejados en la cantidad de tweets por categoría.

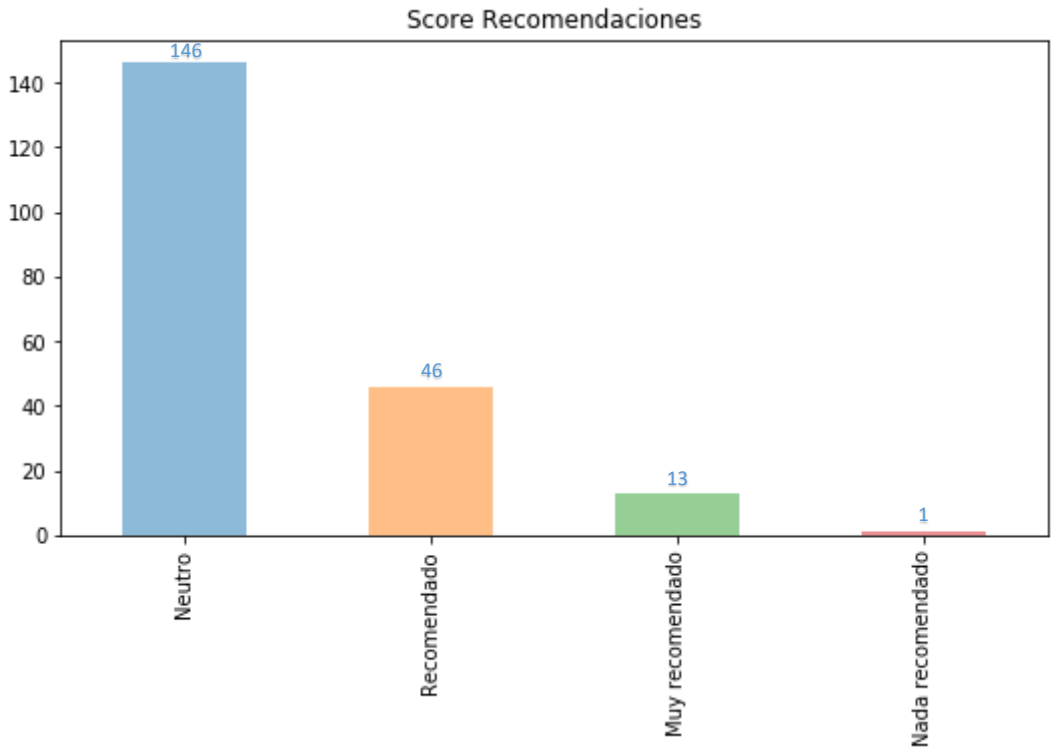


Fig. 5-15 Cantidad de Tweets por categoría WorldThruMyEyes con Aprendizaje Supervisado resumido [Elaboración propia]

Realizando nuevamente el proceso en el corpus del resumen de WorldThruMyEyes; donde se eliminaron todos los tweets repetidos, se volvió a ejecutar las pruebas *K-fold* y los resultados de precisión del algoritmo SVM fueron "más" reales (véase la Tabla 11).

Tabla 11 Resultados del resumen de CrossVal-WorldThruMyEyes con Aprendizaje Supervisado correcto [Elaboración propia]

WorldThruMyEyes c/AS resumido					
	Tweets s/normalizar	Tweets normalizados s/lematizacion	Tweets normalizados c/lematizacion	Bigramas	Trigramas
Precisión y error promedio	0.7393458980044346	0.7488026607538802	0.7488026607538802	0.73847006651188471	0.7339246119733925
	0.74 (+/- 0.12)	0.75 (+/- 0.07)	0.75 (+/- 0.07)	0.74 (+/- 0.05)	0.73 (+/- 0.06)

En la Fig. 5-16 se muestra un ejemplo real de los datos que muestra el programa ejecutado en la consola, que es lo que se observa en la columna 2 de la Tabla 11. Datos que muestran la mayor precisión en el entrenamiento del algo timo SVM y serán tomados en cuenta para la generación del reporte para viajeros.

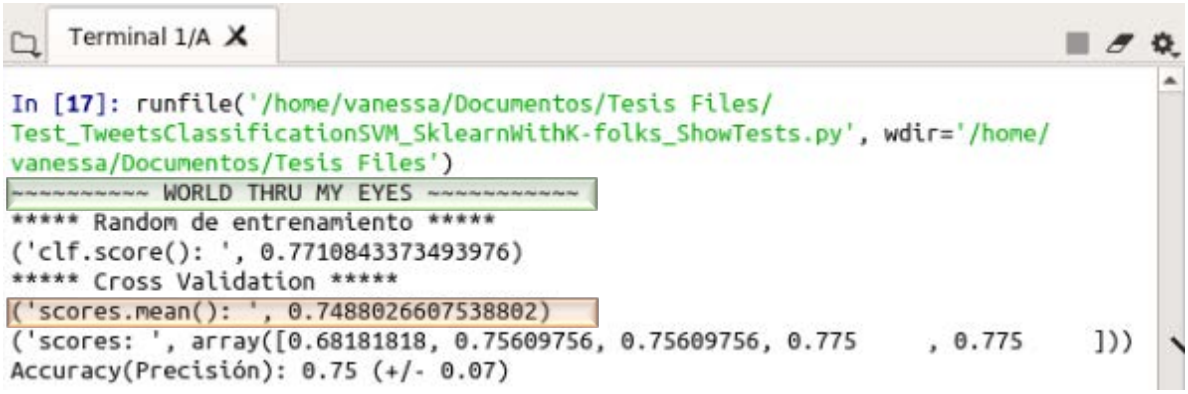


Fig 5-16 Scores del entrenamiento de WoldThruMyEyes desde consola. [Elaboración propia]

Como se puede observar en la Tabla 11 no se llega al objetivo de 0.80 de precisión, esto se debe en parte a que existen varios tweets que muestran su información con el apoyo en imágenes o video; lo que genera que el texto en el tweet sea incompleto y no se encuentre un buen o mal comentario completo en la información de tweet.

Por otro lado, al igual que la cuenta TravelReportMX, la cuenta WorldThruMyEyes dan una mejor precisión de información cuando se normaliza con lematización; por lo tanto, se toma ésta información como base para la generación del reporte ”Reporte para Viajeros MX”.

5.4 REPORTE WEB DE VIAJEROS MX

Después de haber obtenido un score (*Cross-Validation*) considerado como bueno con SVM, el algoritmo dice que el Aprendizaje Supervisado es aceptable y de ahí, se parte para poder mostrar un reporte que ya tiene una buena clasificación (Nada recomendado, No tan recomendado, Neutro, Recomendado, Muy recomendado).

La aplicación de DBpedia ayudó a separar los tweets por estados mexicanos u otros (lugares, países, comida, etc.), para poder permitir al usuario obtener de una manera más organizada la información que se extrajo y fue procesada de usuarios de Twitter. Dando como resultado final, la generación del "Reporte de Viajeros MX", presentado por medio de una página web, como se observa en la Fig. 5-17., donde se seleccionó como **Estado** "Guanajuato" con **Clasificación de Tweet** "recomendado"; mostrando en la columna **Tweet** la información (*tweets*) normalizada.

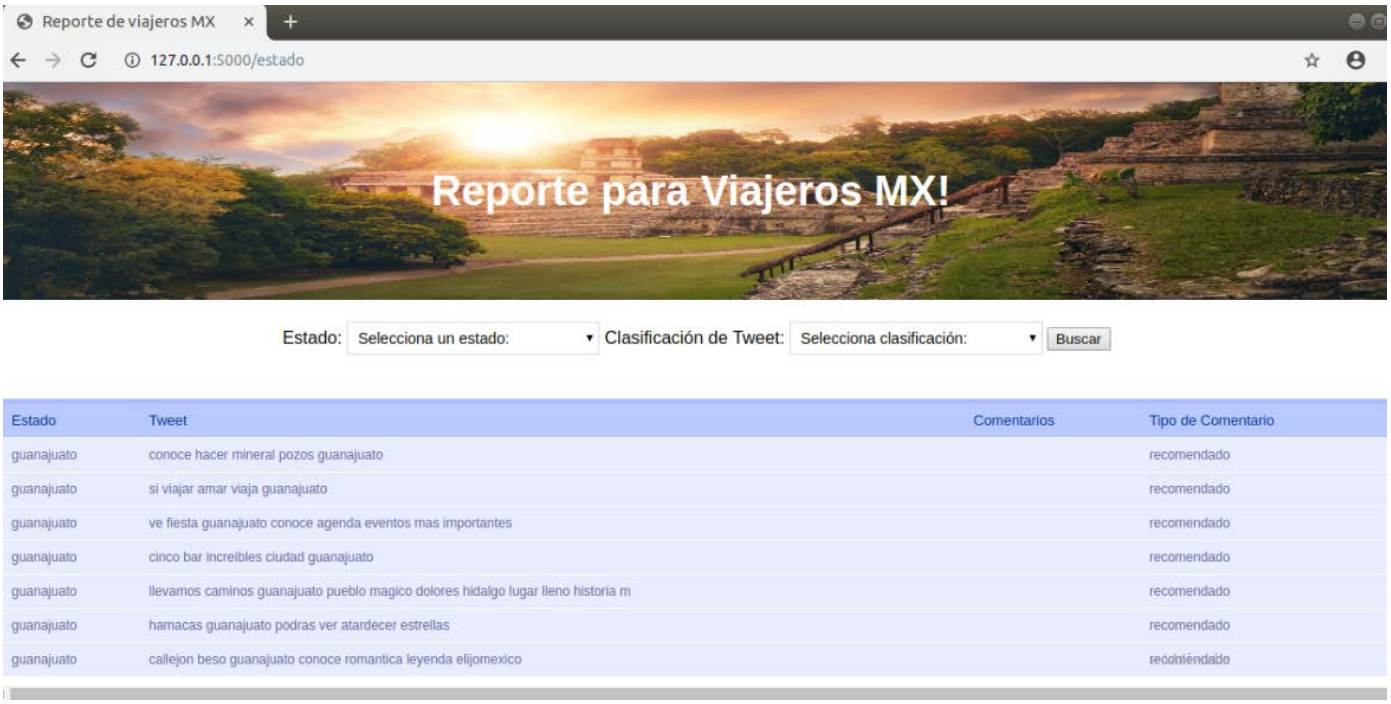


Fig. 5-17 Reporte de Viajeros MX con normalización [Elaboración propia]

6) CAPITULO VI. CONCLUSIONES

La generación del reporte “Reporte para Viajeros MX” fue un proceso largo debido a las características que se eligieron para poder finalizar éste trabajo; las cuales, ayudaron a un mayor aprendizaje tanto en ciencia como en tecnología y, que son el resultado del uso de diferentes herramientas desde el almacenamiento de datos para poder generar un corpus hasta la presentación de la información a viajeros en forma de reporte web.

Dentro de estas herramientas se encuentra el algoritmo SVM. El estudio que se realizó en el estado del arte ayudó a entender que este algoritmo con Aprendizaje Supervisado (AS) es uno de los que ha dado mejores resultados en la clasificación de información, esto debido, a que en conjunto con el AS ayuda al algoritmo a clasificar la información según objetivos deseados. Éste trabajo se enfrascó en tener recomendaciones buenas y malas para viajeros, hecho que no siempre se consiguió en su totalidad en tweets que van acompañados de información útil en imágenes o videos, lo cual hace que el texto esté incompleto y sea clasificado el tweet en el aprendizaje supervisado como neutro; a su vez esto perjudica en los *scores* de la precisión deseada (el objetivo de precisión deseada es 0.80) en las pruebas del algoritmo SVM (como lo fue el caso de las cuentas MexDesconocido y WorldThruMyEyes).

Por otro lado, el uso de 3 cuentas diferentes de Twitter ayudó a tener diferentes datos de viajeros y a poder comparar resultados y comportamientos entre cada una de las cuentas. Por ejemplo, en la cuenta WorldThruMyEyes se detectó que la multiplicidad en un tweet con el mismo AS ayudaba a obtener una precisión de SVM alto, pero a la vez erróneo; ya que al eliminar la multiplicidad de tweets la precisión del algoritmo no dio resultados tan precisos.

A pesar de que el resultado esperado en la precisión del algoritmo SVM no fue la deseada en 2 de las 3 cuentas de Twitter, no estuvo tan alejada de lo esperado; por lo cual tomando los mejores resultados del aprendizaje supervisado que se realizó, se creó el reporte web “Reporte para Viajeros MX” con ayuda del servidor DBpedia (que ayudó a separar los tweets por estados mexicanos u otros y así hacer más fácil la búsqueda de recomendaciones en el reporte) y el micro *framework* Flask (que ayudó a mostrar la información de una forma más amigable a los usuarios finales).

Como trabajos futuros empleando la misma metodología, se podría mejorar éste trabajo no limitando la información en México si no tal vez en los destinos turísticos mayormente recomendados a nivel mundial e incluso obteniendo el corpus de una manera distinta a la que se obtuvo en éste trabajo o tomando en

cuenta el procesamiento de imágenes o video, de modo que se obtenga mayor información de utilidad de las opiniones de viajeros en Twitter.

Finalmente, se considera que el uso de éste trabajo de tesis puede dar como beneficio a los usuarios finales conocer de una forma más rápida las recomendaciones buenas y malas sobre viajes en la república mexicana y así, poder tener un mejor criterio, o un apoyo sobre qué lugares visitar y que hacer en ellos.

BIBLIOGRAFIA

(1) J.C. Ramos, “Detección de acoso en mensajes de Twitter.”, 2017, 35-70.

(2) D. Álvarez, “Identificación de espacios de consenso en redes sociales basado en análisis semántico de producciones lingüísticas.”, 2014, (1,30-39).

(3) J. Selva, “Desarrollo de un sistema de análisis de sentimiento sobre Twitter.”, 2015, 9.

(4) L. Montesinos, “Análisis de sentimientos y predicción de eventos en Twitter.”, 2014, 23-33.

(5) Q. Ye, Z. Zhang, R. Law, “Sentiment classification of online reviews to travel destinations by supervised machine learning approaches.”, 2009, 3-9.

(6) C. D. Manning, H. Schütze, “Foundations of statistical natural language processing (Vol. 999).”, 1999, 46-50.

(7) A. Go, L. Huang, R. Bhayani, “Twitter Sentiment Analysis”, 2009, 2-17.

(8) A. Moreno, C. Pérez, “Lexicon-Based Sentiment Analysis of Twitter Messages in Spanish”, 2013, 93-100.

(9) B. Pang, L. Lee y S. Vaithyanathan, “Sentiment Classification using Machine Learning Techniques”,2002, 1-8.

(10) C. Tauro, S. Aravindh, A. Shreeharsha, “Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases.”, 2012, 1.

(11) Joshua Roesslein. (2019), Tweepy: *Streaming With Tweepy* [Online]. Disponible: http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html, (Obtenida el 28 de septiembre de 2018).

(12) Bowker, L., Pearson, J. “Working with specialized language: a practical guide to using corpora.”, 2002, 1.

(13) J. Yoon, D. Jeong, C. H. Kang, & S. Lee, “Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study.”, 2016, 53-65.

(14) DB-Engines. (2019), DB-Engines Ranking [Online]. Disponible: https://db-engines.com/en/ranking, (Obtenida el 20 de febrero de 2017).

(15) E. Brown. (2017, Enero 19), Collecting/Storing Tweets with Python and MongoDB [Online]. Disponible: http://pythondata.com/collecting-storing-tweets-with-python-and-mongodb., (Recuperada el 21 de marzo de 2017).

(16) E. Tang, Y. Fan, “Performance Comparison between Five NoSQL Databases.”, 2017, 5.

(17) Twitter: Consuming streaming data [Online]. Disponible: https://developer.Twitter.com/en/docs/tutorials/consuming-streaming-data.html, (Recuperada el 28 de septiembre de 2018).

(18)J, Dianes. (2015, Agosto 10), Data Science with Python & R:Sentiment Classification Using Linear Methods [Online]. Disponible: https://www.codementor.io/jadianes/data-science-python-r-sentiment-classification-machine-learning-du107otfg, (Recuperada el 13 de mayo de 2017).

(19)G. Sidorov, Lexicon Redes NL [Online]. Disponible: http://www.cic.ipn.mx/~sidorov/lexicon_redes.txt, (Recuperada el 15 de octubre de 2017)

(20)RANKS NL [Online]. Disponible: https://www.ranks.nl/stopwords/spanish, (Recuperada el 20 de septiembre del 2017)

(21) J. Amat, Máquinas de Vector Soporte (Support Vector Machines, SVMs) [Online]. Disponible: https://rstudio-pubs-static.s3.amazonaws.com/267926_04d64a3e96dc49b4ae6a2ed32fa29fe6.html, (Recuperada el 7 de noviembre de 2018)

(22) Simon Kemp. (2018, Enero 30), DIGITAL IN 2018: WORLD'S INTERNET USERS PASS THE 4 BILLION MARK [Online]. Disponible: https://wearesocial.com/blog/2018/01/global-digital-report-2018, (Recuperada el 7 de mayo de 2018)

(23)Bruno Toledano. (2017, Julio 27), El número de usuarios que ha sumado Twitter en el último trimestre asciende a cero. [Online]. Disponible: https://www.elmundo.es/tecnologia/2017/07/27/5979dc3146163fc6568b4674.html, (Recuperada el 29 de noviembre de 2017)

(24) Gabriel Uccello. (2018, Junio 1), Estadísticas Globales de Twitter 2018 [Online]. Disponible: https://www.flimper.com/blog/es/estadisticas-globales-de-twitter-2018- (Recuperada el 1 de julio del 2018)

(25)Armin Ronacher. (2019), Flask web development, one drop at a time [Online]. Disponible: http://flask.pocoo.org/, (Recuperada el 18 de marzo del 2019)

(26)Malnuer (2016, Enero 5), Analizar Twitter con la Stream API en RStudio [Online]. Disponible: https://malnuer.es/analisis-de-datos/analizar-twitter-con-la-stream-api-en-rstudio/, (Recuperada el 12 de febrero del 2018)

(27)mongoDB (2018), The MongoDB 4.0 Manual [Online]. Disponible: https://docs.mongodb.com/manual, (Recuperada el 21 de febrero del 2017)

(28)K. U. Idehen. (2016, Julio 22), What is DBpedia, and why is it important? [Online]. Disponible: https://medium.com/openlink-software-blog/what-is-dbpedia-and-why-is-it-important-d306b5324f90, (Recuperada en 2018).

GLOSARIO

API: Es la abreviatura de *Application Programming Interface*, es decir, la interfaz permite conectarse programáticamente a una aplicación.

AS: Aprendizaje supervisado.

auscultar: Intentar saber la opinión de una o más personas sobre un asunto.

Cassandra: es una base de datos NoSQL distribuida y basada en un modelo de almacenamiento de «clave-valor», de código abierto que está escrita en Java.

Comments: Texto de ideas o comentarios que los usuarios tienen sobre algún Post.

Gnip: "*Grand Central Station for the social web*" (API de redes sociales de servicios de agregación).

Listener: es una función o método que es suscrito a un evento. Cuando el evento se dispara, el método *listener* obtiene la llamada.

Microblogging: Es una nueva forma de comunicación en Internet que gana adeptos cada día. La magia del *microblogging* es la sintetización en 140 caracteres de lo que quiera contarse, lo cual agiliza la lectura y la comunicación, algo que se ve evolucionar cada día con otros servicios donde se sintetiza el contenido, como resúmenes de libros, píldoras formativas.

NoSQL(no sólo SQL): término que describe las bases de datos no relacionales de alto desempeño. Las bases de datos NoSQL utilizan varios modelos de datos, incluidos los de documentos, gráficos, claves-valores y columnas. Las bases de datos NoSQL son famosas por la facilidad de desarrollo, el desempeño escalable, la alta disponibilidad y la *resiliencia*.

Post: Texto que se publica en un TimeLime.

Resilencia: es la capacidad de sobrevivir.

RDF: Marco de Descripción de Recursos (Resource Description Framework) es una familia de especificaciones de la World Wide Web Consortium originalmente diseñado como un modelo de datos para metadatos.

sharding: Característica de base de datos, se podría traducir como particionado. MongoDB utiliza esta técnica para gestionar la carga de los servidores. Distribuye los datos entre distintos *shards* (conjuntos de servidores que almacenan parte de los datos), para que la carga a la hora de realizar consultas e inserciones se reparta.

Sobreentrenamiento: Es la tendencia que tienen la mayoría de los algoritmos de Machine Learning a ajustarse a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la *función objetivo* que se esta buscando para generalizar.

SPARQL: Acrónimo recursivo del inglés SPARQL (Protocol and RDF Query Language). Se trata de un lenguaje estandarizado para la consulta de grafos RDF Data Access Working Group del World Wide Web Consortium.

Streaming: API de Twitter que se encarga de recibir tweets en tiempo real.

tf·idf: (del inglés Term frequency – Inverse document frequency), frecuencia de término – frecuencia inversa de documento (o sea, la frecuencia de ocurrencia del término en la colección de

documentos), es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección.

Timeline: Pues es la línea en la que se ven todos los tweets tuyos y de las personas a las que sigues, en orden cronológico.

Anexo

STOP WORDS:

STOP WORDS					
un	porque	desde	tuyo	cierto	era
una	por qué	conseguir	ellos	ciertos	eras
unas	estado	consigo	ellas	cierta	eramos
unos	estaba	consigue	nos	ciertas	eran
uno	ante	consigues	nosotros	intentar	modo
sobre	antes	conseguimo	vosotros	intento	bien
todo	siendo	consiguen	vosotras	intenta	cual
también	ambos	ir	si	intentas	cuando
tras	pero	voy	dentro	intentamos	donde
otro	por	va	solo	intentais	mientras
algún	poder	vamos	solamente	intentan	quien
alguno	puede	vais	saber	dos	con
alguna	puedo	van	sabes	bajo	entre
algunos	podemos	vaya	sabe	arriba	sin
algunas	podeis	gueno	sabemos	encima	trabajo
ser	pueden	ha	sabeis	usar	trabajar
es	fui	tener	saben	uso	trabajas
soy	fue	tengo	ultimo	usas	trabaja
eres	fuimos	tiene	largo	usa	trabajamos
somos	fueron	tenemos	bastante	usamos	trabajais
sois	hacer	teneis	haces	usais	trabajan
estoy	hago	tienen	muchos	usan	podria
esta	hace	el	aquellos	emplear	podrias
estamos	hacemos	la	aquellas	empleo	podriamos
estais	haceis	lo	sus	empleas	podrian
estan	hacen	las	entonces	emplean	podriais
como	cada	los	tiempo	ampleamos	yo
en	fin	su	verdad	empleais	aquel
para	incluso	aqui	verdadero	valor	
atras	primero	mio	verdadera	muy	