ISSN 1870-4069

Prototipo de sistema para la clasificación de reseñas de películas

Abdiel Reyes-Vera[1], Francisco Javier Aguilar-Tecuapan[2], Juan Pablo Francisco Posadas-Durán[2], Grigori Sidorov[1]

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

[2] Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatenco, de México, México

abdielreyes81@gmail.com, javiertecuapan@gmail.com, posadas.esimez@gmail.com, sidorov@cic.ipn.mx

Resumen. Clasificar es una tarea que consiste en ordenar información de una manera correcta, si no se tienen los conocimientos necesarios será difícil, por ejemplo, a una persona común si se le dan unos libros de filosofía los clasificaría a su propio entendimiento, lo cual sería muy diferente si fuera una persona docta en el tema, pero aun teniendo los conocimientos, si la cantidad de documentos, libros o textos es grande, una persona no lo podría hacer por sí misma, aun con la ayuda de un grupo de expertos esta tarea sigue siendo larga y fatigante llegando a consumir días, meses, incluso años. En este sentido muchas empresas que ofrecen servicios a través de la web como Netflix, Uber, Amazon, etc. tienen la problemática de manejar comentarios en sus productos o servicios, los cuales, al carecer de una clasificación adecuada, no son tomados en cuenta, además teniendo miles de comentarios por producto hacen que la tarea de revisar los comentarios sea abrumadora. Debido a que estas empresas dependen de la satisfacción del usuario respecto a sus servicios o productos, se ven en la necesidad de revisar los comentarios para tener una retroalimentación, teniendo una mejora continua, por lo que se ven obligados a tener un método para poder clasificarlos eficientemente.

Palabras clave: Clasificación, grandes volúmenes de información, satisfacción del usuario, máquinas de vectores de soporte.

System Prototype for Classification of Movies' Reviews

Abstract. Classifying is a task that consists of ordering information in a correct way, if you don't have the necessary knowledge, it will be difficult, for example, for a person common if you give him some philosophy books he would classify them to his own understanding, which would be very different if he were a learned person on the subject, but even having the knowledge, if the amount of

documents, books or texts is large, a person she could not do it herself, even with the help of a group of experts, this task it is still long and tiring, consuming days, months, even years. In this sense many companies that offer services through the web like Netflix, Uber, Amazon etc. have the problem of managing comments on their products or services, which, lacking an adequate classification, are not taken into account, also having thousands of comments per product make the task of reviewing the comments are overwhelming. Because these companies depend on satisfaction of the user regarding their services or products, they are in need to review the comments to have feedback, having continuous improvement, so they are forced to have a method to classify them efficiently.

Keywords: Classification, large volumes of information, user satisfaction, support vector machines.

1. Introducción

Servicios como Netflix, Crackle, Ebay, Mercado Libre, entre otros, ocupan los comentarios de la gente para poder retroalimentarse, sabiendo que les interesa a sus usuarios pueden mejorar la calidad de sus servicios o productos y a la vez mejorando sus ventas.

Si bien es cierto que estas páginas tienen métodos para categorizar sus artículos, la mayoría de estos métodos son escalas de estrellas de 1-5, donde 5 es el valor más alto, cuando ponemos cantidades cercanas como 2 o 3, hace que cambie la percepción, no permitiendo un juicio más preciso entre un valor y otro, en otros casos poner valores extremos hace pensar que realmente el producto fue una maravilla o una catástrofe, no dejando percibir realmente la opinión del usuario. Para estas empresas es necesario saber que desean u opinan sus clientes, por lo cual tener una aplicación para clasificar los comentarios o reseñas, sería de inmenso valor para las mismas.

Muchas empresas que han dejado de lado la opinión de los usuarios han tenido una tendencia de que vayan a la quiebra, un ejemplo de esto es Televisa, que durante años ignoró la opinión de sus consumidores, actualmente la empresa se encuentra al borde de la quiebra, todo esto por seguir con los viejos modelos, por el otro lado empresas que han escuchado abiertamente las opiniones de sus consumidores han avanzado de forma exponencial. Un ejemplo de lo dicho en el punto anterior fue que en 2004 se estrenó *Lemony Snicket's A Series of Unfortunate Events*, ésta es una serie de libros que fue adaptada a película teniendo un gran éxito, pero debido a problemas internos de la distribuidora que tenía los derechos, no se pudo seguir con las secuelas ya programadas, a partir de esa fecha muchos seguidores de la franquicia hicieron una petición en la página change.org para continuar con la saga, lo cual no ocurrió. No fue sino hasta diez años después que Netflix consideró el relanzar la saga, pero ahora en su propio formato, lanzó una encuesta para saber si el público realmente la deseaba, la respuesta fue masiva y Netflix confirmo la serie, la cual al paso del tiempo se ha convertido en una de las más importantes series hechas por parte de la compañía.

Unos meses atrás se supo de un caso en un Uber, donde el conductor acoso sexualmente de su pasajera, esta persona tenía varias calificaciones negativas, pero solamente estaban representadas por una escala de 1-5, al cometer el acto varias víctimas alzaron la voz en contra del agresor, reconociendo acoso por parte del

conductor a varias mujeres que viajaban solas. Algunas de ellas habían hecho una queja en la aplicación, quejas que estaban en proceso de revisión. Aquí se puede demostrar que el uso de este tipo de clasificación es muy ambiguo e inútil para los tiempos actuales.

2. Trabajos relacionados

El proyecto tomó su principal inspiración en el trabajo del Dr. Maas [1] donde en su trabajo su equipo buscó derivar un modelo probabilístico de documentos que aprende representaciones de palabras. Este componente hace que se no requiera datos etiquetados, y comparte su base con modelos de temas probabilísticos como LDA. Los componentes de sentimiento del modelo utilizan anotaciones de sentimiento para restringir palabras que expresan similares sentimientos y así evitar tener representaciones similares. Esto permite aprender eficientemente los parámetros para el objetivo conjunto función utilizando maximización alterna.

Esta idea que utilizo el doctor fue bien llevada a cabo, pero para propósitos de nuestra investigación optamos por otras opciones, por ejemplo, el trabajo del Dr. Sidorov [3] donde su equipo muestra que al considerar la similitud entre características para el cálculo de similitud de objetos en el Modelo de espacio vectorial (VSM) para algoritmos de aprendizaje automático y otras clases de métodos que implican similitud entre objetos. Aquí se asume que conocemos la similitud entre las características de varias palabras, por lo cual hace innecesario aprender de los datos. Dentro de esto, entendemos que hay palabras que mantienen una correlación, por ejemplo, las palabras comida y comer no significan lo mismo, pero de la misma forma sabemos que tienen cosas en común, son diferentes pero relacionados. Cuando no hay similitud entre las características, nuestra medida de similitud suave es igual a la similitud estándar. Para esto, generalizamos la conocida medida de similitud de coseno en VSM mediante la introducción de lo que llamamos "medida de coseno suave".

Dentro del trabajo se propuso varias fórmulas para el cálculo exacto o aproximado de la medida del coseno suave. Por ejemplo, en uno de ellos se consideró para VSM un nuevo espacio de características que consiste en pares de las características originales ponderadas por su similitud. Nuevamente, para las características que no tienen similitud entre sí, las fórmulas se reducen a la medida estándar del coseno. Los experimentos muestran que la medida suave del coseno proporciona un mejor rendimiento en nuestro estudio de caso: los exámenes de ingreso responden a la tarea de respuesta en CLEF. En estos experimentos, se utilizó n-gramos sintácticos como características y la distancia de Levenshtein como la similitud entre n-gramos, medidos en caracteres o en elementos de n-gramos.

Estos trabajos utilizaron métodos diferentes, por lo cual consideramos prudente utilizar los métodos convencionales.

3. Artefactos propuestos

Basándonos en el problema planteado, se pensó en buscar diversas bases de datos, al final se utilizó la desarrollada por Andrew Maas [1]. Esta base de datos es un conjunto

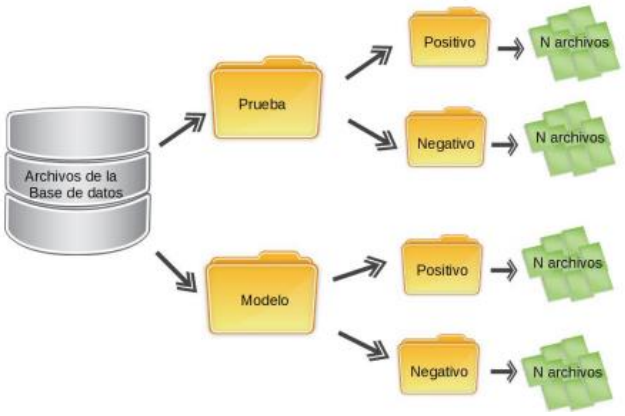

Fig. 1. Estructura de la base de datos.

Tabla 1. Texto antes y después de procesarlo.

Texto sucio	Once again Mr. Costner has dragged out a movie for far longer than necessary. Aside from the terrific sea rescue sequences, of which there are very few I just did not care about any of the characters. Most of us have ghosts in the closet, and Costner's character are realized early on, and then forgotten until much later, by which time I did not care. The character we should really care about is a very cocky, overconfident Ashton Kutcher. The problem is he comes off as kid who thinks he's better than anyone else around him and shows no signs of a cluttered closet.His only obstacle appears to be winning over Costner. Finally when we are well past the half way point of this stinker, Costner tells us all about Kutcher's ghosts. We are told why Kutcher is driven to be the best with no prior inkling or foreshadowing. No magic here, it was all I could do to keep from turning it off an hour in.
Texto limpio	Once again Mr Costner has dragged out movie far longer than necessary Aside the terrific sea rescue sequences which there are very few I just did not care any the characters Most us have ghosts in the closet Costner's character are realized early then forgotten much later which time I did not care The character we should really care is very cocky overconfident Ashton Kutcher The problem is he comes kid who thinks he's better than anyone else around him shows no signs cluttered closet His only obstacle appears be winning Costner Finally we are well past the half way point this stinker Costner tells us all Kutcher's ghosts We are told why Kutcher is driven be the best no prior inkling foreshadowing No magic here it was all 1could do keep turning it hour in

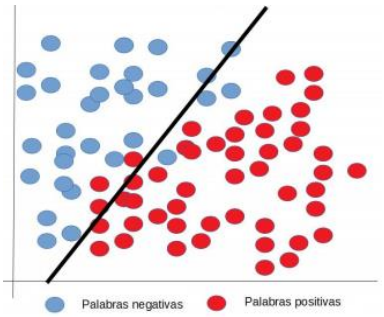

Fig. 2. Representación de las palabras.

de reseñas binarias, esto quiere decir, que le gustó la película o no le gustó a la persona. Además de ser tomadas de la página *MovieDataBase*, sin ser retocadas o corregidas esas reseñas. Ya que, así como las escribió el usuario así se ingresaron a los archivos de la base de datos. En conjunto las reseñas que contienen son 50,000 de las cuales 25,000 son de prueba y 25,000 son de entrenamiento. Estos archivos están distribuidos de maneras iguales para que la carga del modelo esté balanceada.

Generando 2 clases es el método que utilizaremos, ya que utilizaremos el modelo de la base de datos. Generando a su vez dos carpetas, la primera con la información que debe tener los datos, con los que se va a generar el modelo y la segunda con los datos de prueba, que sirven para saber la efectividad del modelo. En estas carpetas se debe tener una muestra del mismo número de archivos tanto en la primera carpeta como en la segunda carpeta, como muestra en la figura 1.

En la etapa se procede a la limpieza de datos, lo que se hace es leer la información con la que desea trabajar, que en nuestro caso es la base de datos, analizando qué palabras son las que no sirven para la creación del modelo. Este análisis se basa en eliminar de la semántica de los textos los signos de puntuación ya que no aportan nada al contenido, también etiquetas *HTML* que vienen en algunos archivos como por ejemplo *
,
, etc* y algunas preposiciones que consideramos poco importantes y que no alteran el sentido o la raíz del mensaje en inglés.

Otras palabras que son eliminadas son las preposiciones ya que afecta muy poco la sintaxis del texto. Se generan versiones depurada de las reseñas (eliminación de etiquetas *HTML*) de la base de datos que se identifican por la extensión pre. Para ilustrar este punto se cita el siguiente texto almacenado en la base de datos con la que se está trabajando y fue sacado al azar. En el siguiente texto que tomamos de la base de datos, se marcará con negritas y se subrayarán las palabras que serán eliminadas en la versión preprocesada. Haciendo hincapié que utilizamos la librería *NLTK*, para toda la limpieza de los archivos, ver Tabla 1.

Los resultados de la metodología usada, constó de 20 experimentos diferentes, de los cuales nos basamos para decir que es la mejor propuesta o forma de tratar un problema similar al nuestro. Siempre y cuando se trate de clasificar datos con máquinas de soporte vectorial con representación lineal que fue el modelo con el mejor accuracy alcanzando. Hicimos una representación de los datos en la siguiente imagen. Ver figura 2.

Tabla 2. Resultados sin lematización.

Método	Lematización	Porcentaje	N-gramas	Tamaño vector
Liblinear	No	88.516	1	(24999, 111684)
Logical regresion	No	86,424	1	(24999, 111684)
Liblinear	No	87,036	2	(24999, 1595609)
Logical regresion	No	84,468	2	(24999, 1595609)
Liblinear	No	81,532	3	(24999, 3396177)
Logical regresion	No	78,392	3	(24999, 3396177)
Liblinear	No	75,8	4	(24999, 3748630)
Logical regresion	No	74,892	4	(24999, 3748630)
Logical regresion	No	69,34	5	(24998, 4431288)
Liblinear	No	69,284	5	(24998, 4431288)

En la línea que los separó se puede observar que nuestro modelo pretende trazar una línea que muestre de manera gráfica cómo los comentarios positivos o negativos pueden ser confundidos y al limpiar de la manera propuesta la tesis se puede compensar este error.

En esta etapa se probó formar diferentes vectores que se iban a procesar, donde variaba la longitud del vector pese a ser el mismo archivo precompilado y el mismo número de archivos tratados. En este caso que fueron de 12,500 archivos positivos y 12,500 negativos. Que se usaron para la creación del modelo matemático y otros 12,500 archivos negativos y positivos respectivamente. Dando un total de 100,000 archivos con aproximadamente 4,431,288 palabras.

Que es una cantidad abismal de texto para leer, ya que un novela consta de 40,000 hasta 90,000 palabras que según [2] es el numero promedio. Dentro de los modelos que se utilizaron fueron *logical regression y liblinear*, todos utilizando el modelo de *bag of word*. Estos vectores de palabras variaron sus parámetros de n-gramas de 1 hasta 5 dando como resultado que mientras mayor fuera el tamaño del n-grama perdía efectividad, ver tabla 2.

Cabe mencionar que el *accuracy* logrado fue mayor que al utilizar la lematización, por lo cual no presentamos esos resultados. Además de mostrar que el vector se hace más chico porque es más probable que se repita una palabra, a una secuencia de palabra, demostrando que entre más pequeño es el vector mejor resultado obtenemos.

Por el otro lado observamos que el hacer *n-gramas* de mayor tamaño, significa aumentar el tamaño del vector, dificultando la coincidencia con otros textos haciendo

que la frecuencia disminuya en los *n-gramas* dando como resultado un valor más burdo. Introduciendo en la máquina de soporte vectorial que exista una línea muy estrecha y mezclando las clases en el hiperplano aumentando el error de las predicciones. De los modelos que se usaron se concluye que el mejor es *Liblinear* por tener mejor efectividad que lógica de regresión con todos los experimentos realizados organizados de mayor efectividad a menor y se presentan qué técnicas se usaron. Poniendo en claro la ventaja de *liblinear* ya que ganó en los primeros cinco puestos de esta tabla usando diferentes técnicas. Ver tabla 3. Esto se debe a la sobre estimación de datos que genera el modelo de análisis de lógica de regresión. Por ende, nuestro prototipo solo cuenta con *liblinear*, *n-gramas* de tamaño 1 y sin lematización.

Tabla 3. Todos los resultados logrados.

Método	Lematizacion	Porcentaje	N-gramas	Tamaño vector
liblinear	no	88,516	1	(24999, 111684)
liblinear	si	88,292	1	(24999, 106814)
liblinear	si	87,14	2	(24999, 1528771)
liblinear	no	87,036	2	(24999, 1595609)
logical regresion	no	86,424	1	(24999, 111684)
logical regreslon	si	86,324	1	(24999, 106814)
logical regresion	si	84,488	2	(24999, 1528771)
logical regresion	no	84,468	2	(24999, 1595609)
liblinear	si	81,74	3	(24999, 3353022)
liblinear	no	81,532	3	(24999, 3396177)
logical regresion	Si	78,392	3	(24999, 3353022)
logical regresion	**no**	78,392	3	(24999, 3396177)
liblinear	Si	75,872	4	(24999, 3737830)
liblinear	no	75,8	4	(24999, 3748630)
logical regresion	si	75,02	4	(24999, 3737830)
logical regreslon	no	74,892	4	(24999, 3748630)
liblinear	si	69,548	5	(24998, 4428725)
logical regresion	si	69,512	5	(24998, 4428725)
logical regresion	no	69,34	5	(24998,4431288)
liblinear	no	69.284	5	(24998, 4431288)

Respecto a otros trabajos que podemos comparar seria el mismo hecho por Maas [1] que dentro de sus resultados obtuvo un resultado menor que el nuestro en varios casos, por ejemplo, Maas en su mejor resultado obtuvo 88.89 eso sin utilizar las etiquetas hechas por él, y con las etiquetas 88.33, ya que nuestro proyecto es el de clasificar las películas comparamos nuestro resultado respecto a este último *accuracy*, ya que nosotros ganamos por 88.55%.

4. Conclusiones y trabajo a futuro

Se logró un porcentaje de predicción mayor, que la lograda por el autor de la base de datos de 88.33 por ciento en el 2010. Al agregar una función que elimina puntos, comas, etiquetas HTML, algunas preposiciones, etc. Además de tratar de una manera convencional la bolsa de palabras, ya que el autor trataba de predecir posibilidades de palabras subsecuentes a la anterior generando ruido al sistema de clasificación. Además de ver el potencial que tiene este algoritmo ya que pueden ser usados otros idiomas como el español, el portugués o que tengan una estructura similar al inglés. Encontramos en este sistema de sugerencias una utilidad para trabajar con datos de gran tamaño y de manera rápida. Pudiendo dar soluciones en áreas de ventas, marketing, industrias, etc. En la minería de datos y análisis de la información de lo que opina un usuario de su producto o servicio. Haciendo más fácil la tarea de evaluar esa información, que resulta ser importantes para la mejora y creación de productos para los usuarios, por lo que muchas empresas pagan jugosas cantidades de dinero.

Como trabajo futuro, se plantea lo siguiente:

- Crear una base de datos en español, para que nuestro sistema sirva en español y no solamente en inglés.
- Compilar un conjunto de datos de prueba, para evaluar nuestro sistema en dicho idioma.
- Implementar un módulo que descargue información de la web para compilar bases de datos sobre reseñas.
- Probar otros clasificadores.
- Probar otras combinaciones de representaciones.
- Probar un diccionario de emoticones en diversos idiomas.

Agradecimientos. Agradecemos a CONACYT, SNI, IPN (SIP, COFAA), apoyo de proyectos SIP 20200859 y 20200797 y Conacyt A1-S-47854.

Referencias

1. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. 49th Annual Meeting of the Association for Computational Linguistics (ACL) (2011)
2. Freixinet, L.J.: Cuántas palabras tiene una novela. https://www.cafedelescritor.com/cuantas-palabras-tiene-una-novela/ (2020)
3. Sidorov, G., Gelbukh, A., Gómez-Adorno, H., Pinto, D.: Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas, 18(3), pp. 491–504 (2014)