ISBN 978 -987-633-574-4

Técnicas de Análisis de Sentimientos Aplicadas a la Valoración de Opiniones en el Lenguaje Español

Germán Rosenbrock[1], Sebastián Trossero[1], Andrés Pascal[1,2]

1 Fac. de Ciencia y Tecnología, Univ. Autónoma de Entre Ríos, Ruta 11 - Km. 11, Oro Verde, Entre Ríos, Argentina.
2 Fac. Regional Concepción del Uruguay, U.T.N, Ing Pereira 676, Concepción del Uruguay, Entre Ríos, Argentina.
rosenbrock.german@uader.edu.ar, trossero.sebastian@uader.edu.ar, andrespascal22@gmail.com

Abstract. En el presente existen grandes cantidades de datos en formato de texto escritos en el lenguaje natural, disponibles principalmente en sitios web y redes sociales, que crece día a día. El análisis manual de estos volúmenes de información es actualmente impráctico y costoso, por lo cual se hace necesario el uso de técnicas automatizadas para su procesamiento y análisis. La Minería de Opinión o Análisis de Sentimientos estudia la extracción de información a partir de datos subjetivos y es relativamente reciente. En los últimos años se han propuesto varios modelos de procesamiento del lenguaje natural para resolver el problema particular de clasificación de sentimientos. En este trabajo examinamos el rendimiento de varios de estos modelos aplicados a un caso donde los textos están escritos en el lenguaje castellano coloquial, lo que representa un desafío adicional. El caso propuesto es un conjunto de más de 50.000 reseñas de películas, extraídas del sitio www.cinesargentinos.com.ar.

Palabras claves: Minería de opinión, Análisis de sentimientos, Procesamiento del lenguaje natural en español, Data Mining, Análisis subjetivo.

1 Introducción

En un proceso de toma de decisiones, es fundamental contar con información oportuna, confiable y completa que permita un análisis real de la situación. En ciertos casos, los datos de origen son opiniones personales. En forma previa a la Web 2.0, su importancia no era alta debido a la escasa cantidad de textos que registraban opiniones. En el presente, con la disponibilidad masiva de este tipo de información, surgen nuevas oportunidades y desafíos en la búsqueda, comprensión e interpretación de la misma. Sin embargo, la búsqueda en estos sitios y la posterior valoración de las opiniones en forma manual es un trabajo intenso y costoso, por lo que es necesario contar con sistemas que automaticen este proceso.

El Análisis de Sentimientos o Minería de Opiniones estudia la interpretación automática de opiniones y sentimientos expresados mediante el lenguaje natural. Es utilizada por organizaciones, por ejemplo, para el análisis de su imagen o para determinar necesidades o también el grado de aceptación de nuevos productos. La literatura, además, muestra varios otros tipos de aplicaciones, incluyendo: valoración de películas [1], opiniones sobre deportes [2], turismo [3, 4], política [5], educación [6], salud [7], finanzas [8] y automóviles [9].

ISBN 978 -987-633-574-4

Este trabajo presenta la aplicación y comparación de distintas técnicas de aprendizaje automático como Máquinas de Vectores de Soporte (SVM), Clasificador Bayesiano Ingenuo (Naïve-Bayes), Máxima Entropía y Random Forest, con el enfoque clásico de bolsa de palabras, contra técnicas más actuales como la utilización de embeddings con redes neuronales recurrentes y Transformers, también conocidos como Modelos de Lenguaje. El caso de estudio se realiza sobre los comentarios y valoraciones de usuarios acerca de películas extraídas del sitio www.cinesargentinos.com.ar. La selección de este sitio se realizó teniendo como criterio la disponibilidad de los datos, la cantidad de opiniones, el nivel de informalidad en el uso del lenguaje, la disponibilidad de una valoración ya registrada para cada opinión (puntuaciones por estrellas), y la existencia de distintos aspectos a evaluar por cada opinión.

2 Marco Teórico

El análisis del sentimiento o la minería de opinión es el estudio computacional de opiniones, sentimientos y emociones expresadas a través de un texto. En general, las opiniones pueden centrarse en un producto, un servicio, un individuo, una organización, un evento o un tema. Utilizamos el término objeto para denotar la entidad de destino que se ha comentado. Un objeto puede además tener un conjunto de componentes (o partes) y un conjunto de atributos o propiedades). Cada componente puede tener sus propios subcomponentes y su conjunto de atributos, y así sucesivamente.

Lui [10] formaliza estos conceptos mediante las siguientes definiciones:

- **Objeto**: un objeto *o* es una entidad que puede ser un producto, persona, evento, organización o tema. Está asociado a un par, *o: (T, A)*, donde *T* es una jerarquía de componentes (o partes) y *A* es un conjunto de atributos de *o*. Cada componente tiene su propio conjunto de componentes y atributos.
- **Opinión**: una opinión sobre una característica *f* es una actitud, emoción o valoración positiva o negativa sobre *f* .
- **Orientación de una opinión**: la orientación de una opinión sobre una característica *f* indica si la opinión es positiva, negativa o neutral.

Asimismo, una opinión puede ser directa (respecto a un único objeto), o bien comparativa, que expresa una relación de similitudes, diferencias y/o preferencias entre dos o más objetos emitida por el titular de opinión sobre algunas de las características compartidas entre los objetos.

Nuestro problema es establecer si un documento expresa una opinión positiva o negativa de un objeto, aplicando diferentes técnicas de evaluación de opiniones sobre una misma base de datos, para analizar sus desempeños en forma comparativa.

Los métodos seleccionados para nuestro estudio son todos de aprendizaje supervisado, lo que significa que se requiere conocer la clase a la que pertenece la observación al momento de su entrenamiento. Los métodos son Naive Bayes, Random Forest, Regresión Logística y SVM con la representación clásica de bolsa de palabras; Redes Neuronales Recurrentes con el embedding Word2Vec y por último, para la arquitectura de Transformers se utilizó el modelo de lenguaje BETO, una versión en español del modelo original BERT.

A continuación se realiza una breve descripción de cada una de estas técnicas.

2.1 Naïve Bayes

Este algoritmo de clasificación se basa en el *Teorema de Bayes* de probabilidad condicional, además supone la independencia entre las variables predictoras. Ya que en muchos casos esta independencia no es real, se lo denomina 'Naïve' o 'Ingenuo' [11, 12, 13, 14]. La clasificación que realiza este método está dada por la probabilidad de que una observación pertenezca a una clase, dadas las probabilidades de sus variables predictoras. Es la técnica más utilizada como base de comparación.

2.2 Random Forest

Es un clasificador que consiste en un ensamble de múltiples árboles de decisión [15]. Cada uno de estos árboles se entrena con un subconjunto de registros y un subconjunto de variables del conjunto de datos tomados de forma aleatoria.

Este algoritmo puede manejar conjuntos de datos de gran dimensionalidad sin verse afectado por la colinealidad. Otra cualidad que posee este algoritmo es que se puede obtener como salida la importancia de las variables, es decir, las que más influyen en el modelo.

Es difícil de interpretar, ya que es un modelo de caja negra y dependiendo de los parámetros utilizados, en algunos casos se puede caer en overfitting.

2.3 Regresión Logística

La Regresión Logística (también conocido como clasificador de máxima entropía) [16, 17, 18], es un modelo matemático utilizado para predecir el resultado de una variable categórica, por lo general dicotómica, en función de las variables independientes o predictoras. La predicción que se obtiene es la probabilidad de pertenecer a cada clase.

Una de las ventajas fundamentales de la regresión logística sobre otras técnicas, es que el resultado del modelo entrenado se puede interpretar fácilmente. Esto se debe a que el coeficiente obtenido para cada variable dependiente, indica de qué manera influye en el modelo dicha variable. Otras ventajas son su simplicidad y eficacia.

2.4 SVM

SVM (Support Vector Machine) [2, 12, 13, 19], es un algoritmo de clasificación binario, que consiste en encontrar un hiperplano que maximice la separación entre las clases. SVM se puede utilizar con diferentes kernels dependiendo si los datos son linealmente separables o no, lo cual es un parámetro a definir. El entrenamiento de SVM con grandes conjuntos de datos no es recomendable porque no es muy eficiente.

2.5 Word2Vec+LSTM

Los *word embeddings* son una forma de representación de palabras de un documento, que además de representar la palabra aporta información de contexto dentro del documento y de similaridad con otras palabras. Word2Vec es una técnica de word embedding desarrollada en 2013 por Mikolov [20] que utiliza como representación de palabras un vector multidimensional. De esta forma, las palabras relacionadas o similares se encuentran en zonas cercanas dentro de esta representación. Estos

ISBN 978 -987-633-574-4

vectores se utilizan luego como entrada de redes neuronales para realizar tareas como clasificación, traducción o resumen de textos [21, 22].

Las redes neuronales recurrentes LSTM (Long Short Term Memory) tienen la capacidad de persistir información de estados anteriores para calcular los siguientes estados. Es por eso que son muy útiles para trabajar con secuencias, como por ejemplo en modelos de procesamiento del lenguaje natural, ya que se trata de secuencia de palabras. La limitación que tienen es que esa capacidad de "recordar" estados previos es a corto plazo. Las LSTM en cambio, son un tipo de redes neurales recurrentes que tienen ese mismo comportamiento pero a más largo plazo [23].

2.6 BERT

A finales de 2017 Google presenta una nueva arquitectura denominada Transformer [24] en la cual propone quitar las capas recurrentes y convolucionales de las redes utilizadas hasta el momento, a cambio de mecanismos o capas de atención. Estas capas de atención codifican las palabras en función de las demás palabras de la frase, permitiendo introducir información del contexto junto con la representación de cada palabra.

BERT (Bidirectional Encoder Representations from Transformers) [25] es un Modelo de Lenguaje diseñado para entrenar representaciones bidireccionales profundas a partir de textos sin etiquetar, tomando en cuenta tanto el contexto izquierdo como derecho en todas las capas. BERT ha sido pre-entrenado mediante aprendizaje no supervisado a partir de corpus de gran tamaño en idioma inglés. A diferencia de los modelos secuenciales o recurrentes tradicionales, la arquitectura de atención procesa toda la secuencia de entrada a la vez, permitiendo que todos los tokens de entrada se procesen en paralelo.

Para superar su limitación inicial de funcionamiento sólo para el inglés, han surgido versiones que soportan distintos lenguajes, o inclusive múltiples lenguajes en uno, como es el caso de mBERT [26]. Para el lenguaje español en particular, uno de los modelos más conocidos se llama BETO [27] y tiene las mismas características antes mencionadas de BERT, pero con la diferencia que el pre-entrenamiento se realizó con textos en español.

3 Experimentos Realizados

3.1 Conjunto de datos

Este estudio fue realizado sobre una base de datos de comentarios extraídos del sitio web www.cinesagentinos.com.ar; los comentarios son reseñas de distintas películas que los usuarios aportan sin ninguna estructura definida, donde además se pondera la película con un puntaje de una a cinco estrellas. Se definió que un comentario se clasifica como "positivo" si posee cuatro estrellas o más. El lote de datos final fue de 52.309 comentarios de los cuales 36.661 fueron etiquetados como positivos (aproximadamente el 70%).

ISBN 978 -987-633-574-4

3.2 Métricas utilizadas

Para evaluar la capacidad predictiva de los modelos se utilizaron las métricas usuales para estos casos de estudio, definidas de la siguiente manera:

$$Accuracy = (TP+TN) / (TP+FP+TN+FN)$$
$$Precission = TP / (TP+FP)$$
$$Recall = TP/(TP+FN)$$
$$F1\_score = 2 * (Precision* Recall) / (Precision+Recall)$$

donde: TP=True Positive, TN=True Negative, FP=False Positive, FN=False Negative.

3.3 Descripción de los experimentos

Con el fin de obtener el mejor modelo para cada uno de los algoritmos se realizó una búsqueda de hiperparámetros por medio del método Grid Search, entrenando modelos con distintos valores de los parámetros propios de cada algoritmo, quitando o dejando las “stop words” y con distintas cantidades de las palabras más frecuentes en los comentarios. A continuación se describen los hiperparámetros de ajuste:

- Naïve Bayes: ajusta un parámetro “Alpha” entre 0 (cero) y 1 (uno); es un parámetro de corrección o regularización para evitar problemas con la probabilidad cero de eventos ocultos.
- Random Forest: se define la cantidad de estimadores que corresponde a la cantidad de árboles de decisión que se utilizan. Los valores posibles van desde 1 en adelante, sin un límite superior.
- Regresión logística: se ajusta un parámetro llamado Solver con los posibles valores: "liblinear","sag" y "saga". Cada uno ajusta el modelo tomando distintas métricas de penalización.
- SVM: en el caso de este algoritmo se define el tipo de Kernel que utiliza; los posibles valores son: linear, polynomial y RBF.

Como representación del texto de entrada, para los cuatro primeros algoritmos se utilizaron “Bolsas de palabras” (en adelante BdP), que se definen mediante vectores cuyas columnas están indexadas por cada una de las palabras que se encuentran en el conjunto de datos completo, y que almacena en sus valores la concurrencia de esas palabras en el comentario. A este método también se ajustaron los siguientes parámetros para el Grid Search:

- La cantidad de palabras (o columnas de los vectores): se define *n* como la cantidad máxima de palabras a utilizar en la BdP, teniendo en cuenta que sean las *n* palabras con mayor concurrencia en el conjunto de datos. Este parámetro fue ajustado entre 1.000 y 50.000 palabras.
- Eliminar Stop Words: las Stop Words (en adelante SW) son palabras del lenguaje que no poseen riqueza semántica, por ejemplo, los conectores. En los experimentos se utilizaron dos diccionarios distintos de SW para el lenguaje español, uno incluido en la librería NLTK y el otro generado a partir del mismo. Los valores de ajuste de este hiperparámetro fueron: “No borrar SW”, “Borrar diccionario completo” y “Borrar diccionario alternativo”.

ISBN 978 -987-633-574-4

Para el caso de word2vec, el embedding fue generado a partir de las palabras de los mismos comentarios; mientras que para el modelo de Transformers, BETO ya cuenta con un embedding pre-entrenado con palabras en español.

Para cada iteración de parámetros de Grid Search se entrenaron cinco modelos distintos utilizando la técnica Monte Carlo Cross Validation. Las divisiones del conjunto de datos para entrenamiento y prueba se realizaron al 80% y 20% respectivamente. Se calcularon las métricas *Accuracy*, *Presicion*, *Recall* y *F1-Score*, tomando esta última como referencia para determinar el mejor modelo y seleccionar sus hiperparámetros como los óptimos.

3.4 Resultados

Los resultados obtenidos por los distintos algoritmos se muestran en la Tabla 1. A continuación se realiza una breve descripción de los mismos, y los hiperparámetros con los que se obtuvieron los mejores valores.

Tabla 1 Puntajes de los modelos de clasificación sobre el conjunto de datos de prueba.

Modelos	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	0.80	0.80	0.81	0.81
Random Forest	0.80	0.82	0.77	0.79
Regresión logística	0.80	0.80	0.80	0.80
SVM	0.79	0.79	0.79	0.79
LSTM + Word2Vec	0.81	0.85	0.88	0.87
BERT (BETO)	0.83	0.85	0.91	0.88

3.4.1 Naïve Bayes

El resultado de la búsqueda Grid Search para este algoritmo obtuvo el mejor F1-score de los cuatro primeros algoritmos con una ponderación aproximada del 81% de clasificación correcta (ver Tabla 1). El modelo fue entrenado con un valor 1 en el parámetro Alpha y 50.000 palabras en la BdP sin eliminar ninguna “Stop Word”.

Los distintos experimentos realizados arrojan resultados que demuestran que para esta aplicación el aumento del parámetro Alpha también aumenta la potencia predictiva del modelo resultante. Del mismo modo se observa que cuantas más palabras se utilicen para entrenar el modelo lleva a un aumento del F1-score. Por otro lado, la eliminación de SW no tiene resultados positivos en cuanto al F1-score, por el contrario, no eliminarlas mejora los resultados un 1%.

3.4.2 Random Forest

En el caso de este algoritmo se realizaron búsquedas de Grid Search ampliando la cantidad de estimadores hasta que el aumento de los puntajes no fue significativo. El modelo óptimo lo encontramos con 2.000 estimadores y una BdP de 50.000 palabras,

ISBN 978 -987-633-574-4

sin quitar las SWs. El porcentaje de comentarios correctamente clasificados por este modelo fue 79%.

3.4.3 Regresión Logística

El “Solver” que maximizó el F1-Score para este problema fue “saga” con un puntaje aproximado de 80%. En este caso la cantidad óptima de palabras fueron 40.000 para conformar la BdP, al igual que los otros, sin quitar SWs.

3.4.4 SVM

Este algoritmo se optimizó con el Kernel “linear” con 1.000 palabras en su BdP sin quitar las SWs tampoco. De los cuatro modelos que emplearon BdP, fue el que obtuvo el puntaje más bajo de F1-Score, con aproximadamente el 79% de la clasificación correcta.

3.4.5 LSTM+Word2vec

El mejor resultado obtenido fue generando un embedding de 500 palabras, sin quitar SWs, y con un learning rate de 0,02 en el entrenamiento de la red neuronal. Se obtuvo un F1-Score de 87%.

3.4.6 BETO (BERT)

El F1-score obtenido en este experimento fue de 88%, clasificando de forma incorrecta solo 1.815 del total de 10.462 comentarios. La tasa de *learning rate* óptima fue de 0,03, el *batch size* de 64, y la cantidad de palabras seleccionadas por comentario fue 150. Tampoco se quitaron las SWs.

3.5 Análisis de los Resultados

Tal como se esperaba, las dos técnicas más recientes obtuvieron los mejores resultados, alrededor de un 7% más que las primeras cuatro, aunque entre ellas no hay diferencias significativa en este caso. En cuanto al preproceso de los datos se observó que la eliminación de Stop Words tanto del diccionario original de la librería NLTK como el modificado, no generó mejores resultados si no que, por el contrario, disminuyó su rendimiento.

En la literatura reciente, existen distintos trabajos de clasificación de comentarios de películas escritos en inglés [28, 29, 30], en donde utilizando BERT se obtuvieron como resultado entre un 85% y un 94% de Accuracy, mientras que en nuestro caso de estudio el valor alcanzado fue 83%, es decir, entre un 2% y un 11% menos. Esta diferencia puede tener varias causas: diferencias propias del lenguaje, pre-entrenamiento con un corpus de menor tamaño, diferencias en el nivel de informalidad del lenguaje coloquial utilizado, o incluso, mejor ajuste de algunos hiperparámetros.

3.5.1 Comentarios mal clasificados

Para comparar los comentarios mal clasificados tomamos en cuenta solo los 2 mejores modelos obtenidos, LSTM+Word2vec y BETO. Del total de 10.462 comentarios del conjunto de testing, 1.992 fueron mal clasificados utilizando el primer algoritmo, mientras que con BERT fueron 1.815. Teniendo en cuenta que se utilizó el mismo conjunto de testing para los experimentos, se observó que 994 comentarios fueron mal clasificados por ambos algoritmos a la vez.

3.5.2 Causas de la clasificación errónea

Analizando los comentarios mal clasificados, encontramos al menos cinco posibles causas por las cuales el comentario no obtuvo la clasificación correcta:

1. Casos en los que, a pesar de que el comentario tiene una connotación positiva, la etiqueta original del mismo es negativa. Es decir, el autor del comentario escribió una opinión positiva de la película, pero la calificó negativamente.
   Por ejemplo: *"la película me pareció buena, mantiene el suspenso y está muy bien filmada, el efecto 3d está muy bien logrado"*, *"comedia entretenida, divertida, para pasar un buen rato y reírse bastante. Cameron Díaz es muy buena en la comedia y el elenco está muy bien" o "linda comedia, buenas actuaciones y los actores se complementan muy bien pero lo mejor de la película en mi opinión es la elección de música, el mejor soundtrack que he visto en mucho tiempo"*.
2. Casos de comentarios calificados positivamente por el usuario, pero acompañado de un comentario con mensaje negativo: *"No me terminó de convencer. A la peli le pasa factura todos los problemas que tuvo a la hora de realizarse. La trama a pesar de ser interesante se hace por momentos algo aburrida." o "Decepcionante. Se nota que le falta media hora. Para pasar el rato pero nada más. Está hecha sin ganas"*
3. Comentarios ambiguos, es decir, con cierto balance entre lo positivo y negativo. Por ejemplo, *"supero mis expectativas, las escenas de susto un poco predecibles" o " los primeros minutos son algo aburridos pero al pasar los minutos la pelicula es cada vez es entretenida"*.
4. Frases con sentido figurado, que probablemente no son aprendidas correctamente por el modelo: *"se paso en un suspiro", "Navegando aguas misteriosas debería ser la frase de esta saga" o "sin tramos de baches"*.
5. Negación y a veces doble o triple negación en la misma frase: es probable que los modelos tengan problemas cuando se invierte el sentido de una frase a través de la negación: "no es una pelicula de la que te arrepientas de haber visto" o "Esta nueva entrega no aporta ni suma nada".

Los primeros dos casos no están asociados a los modelos sino a los datos, y sólo son problemáticos cuando el entrenamiento se realiza sobre un corpus que posee una cantidad significativa de ellos.

Respecto a los comentarios ambiguos, una solución parcial que se presenta en distintos trabajos, es definir una tercer clase "neutral" para los casos en los cuales no está claro si el comentarios es positivo o negativo. Las últimas dos causas son

conocidas limitaciones de la mayoría de los modelos, ya que hasta el momento ningún modelo comprende realmente el significado del texto, sino que se basan en las relaciones de co-ocurrencia que encuentran entre las palabras.

4 Conclusiones y Trabajo Futuro

En este trabajo se presenta la aplicación, búsqueda de hiperparámetros, comparación y análisis de resultados de distintas técnicas de aprendizaje automático utilizadas para el Procesamiento del Lenguaje Natural. El caso de estudio fue un conjunto de más de 50.000 comentarios en lenguaje español coloquial sobre películas, extraídos del sitio www.cinesargentinos.com.ar. Los resultados indican que las técnicas más nuevas, Word2vec+LSTM y BERT, son superiores a los modelos anteriores, aunque los porcentajes de acierto obtenidos en este estudio son menores que los publicados sobre casos similares en los cuales los textos se encuentran en idioma inglés.

Algunas de las tareas que se plantean como trabajo futuro son:

- Re-etiquetar los comentarios mal etiquetados del conjunto de datos y volver a ejecutar los experimentos.
- Realizar un ajuste fino del modelo BETO, utilizando un porcentaje de los comentarios como conjunto de entrenamiento.
- Agregar una clase “neutra” en los procesos de entrenamiento y clasificación.
- Discriminar entre frases con sentido literal y figurado, y entrenar clasificadores separados para cada caso.

Referencias

1. KuatYessenov. Sentiment Analysis of Movie Review Comments. 2009.
2. N. LI and D. D. W. Using text mining and sentiment analysis for online forums hotspot detection and forecast. DecisionSupportSystems, vol. 48, nº 2, pp. 354 - 368, 2010.
3. L. C. Fiol, J. S. García, M. M. T. Miguel and S. F. Coll, «La importancia de las comunidades virtuales para el análisis del valor de marca. El caso de TripAdvisor en Hong Kong y París,» Papers de turisme, nº 52, pp. 89-115, 2012.
4. C. Henriquez, J. Guzmán and D. Salcedo. Minería de Opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles. Procesamiento del Lenguaje Natural, vol. 56, pp. 25-32., 2016.
5. S. Rill, D. Reinel, J. Scheidt and R. V. Zicari. PoliTwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis. Knowledge-Based Systems, vol. 69, pp. 24-33, 2014.
6. A. Ortigosa, J. M. Martín and R. M. Carro. Sentiment analysis in Facebook and its application to e-learning. Computers in Human Behavior, vol. 31, pp. 527-541, 2014.
7. F. Greaves, D. Ramirez-Cano, C. Millett, A. Darzi and L. Donaldson. Use of Sentiment Analysis for Capturing Patient Experience From Free-Text Comments Posted Online. Journal of medical Internet research, vol. 15, nº 11, 2013.
8. X. Dong, Q. Zou and Y. Guan. Set-Similarity joins based semi-supervised sentiment analysis. Neural Information Processing. Springer Berlin Heidelberg, 2012., from Neural Information Processing, Springer Berlin Heidelberg, 2012, pp. 176-183.
9. P. D. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th annual meeting on association for computational linguistics, Stroudsburg, PA, USA, 2002.

10. Liu B., Zhang L. (2012) A Survey of Opinion Mining and Sentiment Analysis. In: Aggarwal C., Zhai C. (eds) Mining Text Data. Springer, Boston, MA.
11. N. LI and D. D. W. Using text mining and sentiment analysis for online forums hotspot detection and forecast. Decision Support Systems, vol. 48, nº 2, pp. 354 - 368, 2010.
12. A. Abbasi, H. Chen and A. Salem. Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. ACM Transactions on Information Systems (TOIS), vol. 26, nº 3, p. 12, 2008.
13. F. Pla and L.-F. Hurtado. Sentiment Analysis in Twitter for Spanish. Natural Language Processing and Information Systems, pp. 208 - 213, 2014.
14. Gutiérrez Esparza Guadalupe, Margain Fuentes María de Lourdes, Ramírez del Real Tania Aglaé, Canul Reich, Juana, Un modelo basado en el Clasificador Naïve Bayes para la evaluación del desempeño docente, RIED. Revista Iberoamericana de Educación a Distancia (volumen: 20, núm. 2) pp. 293 – 313, 2017.
15. Belgiu M., Dragut L. Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, Volume 114, 2016.
16. Pang, Bo & Lee, Lillian & Vaithyanathan, Shivakumar. (2002). Thumbs up? Sentiment Classification Using Machine Learning Techniques. EMNLP.
17. Wang, Z. (2010). Document Classification Algorithm Based on Kernel Logistic Regression. Industrial and Information Systems (IIS), 2010 2nd International Conference on (Volume: 1) (págs. 76 - 79). Dalian: IEEE.
18. Kamran Kowsari, kiana Jafari Meimandi. Text Classification Algorithms: A Survey, 2019, Information Open Access Journals.
19. David Meyer, Support Vector Machines, The Interface to libsvm in package e1071, FH Technikum Wien, Austria, 2019.
20. T. Mikolov, I. Sutskever, K. Chen, et al., Distributed Representations of Words and Phrases and their Compositionality, arxiv:1310.4546v1, 2013.
21. A. Aubaid y A. Mishra, Text Classification Using Word Embedding in Rule-Based Methodologies: A Systematic Mapping, TEM Journal. Volume 7, Issue 4, Pages 902-914, ISSN 2217-8309, 2018.
22. T. López Solaz, J. Troyano, J. Ortega y F. Enríquez, Una aproximación al uso de word embeddings en una tarea de similitud de textos en espa˜nol, Procesamiento del Lenguaje Natural, Revista n° 57, pág. 67-74, 2016.
23. T. Sainath, O. Vinyals, A. Senior y H. Sak, Convolutional, long short-term memory, fully connected deep neural networks, 2015.
24. A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need. 2017.
25. J. Devlin, M. Chang, K. Lee y K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2, 2019.
26. T. Pires, E. Schlinger y D. Garrette, How multilingual is Multilingual BERT?, arXiv:1906.01502v1, 2019.
27. J. Cañete, G. Chaperon, R. Fuentes and J. Ho, Spanish Pre-Trained BERT Model and Evaluation Data, PML4DC at ICLR 2020, 2020.
28. M. Munikar, S. Shakya and A. Shrestha, Fine-grained Sentiment Classification using BERT, arXiv:1910.03474v1, 2019.
29. L. Maltoudoglou, A. Paisios, H. Papadopoulos, BERT-based Conformal Predictor for Sentiment Analysis, Proceedings of Machine Learning Research 128:1–16, 2020.
30. S. Garg and G. Ramakrishnan, BAE: BERT-based Adversarial Examples for Text Classification, arXiv:2004.01970v3, 2020.