

Clasificación automática de la orientación semántica de opiniones mediante características lingüísticas

Alonso Palomino Garibay y Sofia N. Galicia-Haro

Facultad de Ciencias, UNAM, México, D.F.

alonsop@ciencias.unam.mx, sng@ciencias.unam.mx

Resumen. En este trabajo examinamos el problema de clasificar la orientación semántica de opiniones de productos comerciales, en idioma Español. Exploramos las características de la colección de opiniones extraídas de Internet y estudiamos el uso de bigramas de afirmación como características de entrenamiento para un método supervisado. Analizamos la combinación de estas características y medimos su desempeño al utilizarse en el método de Máquinas de soporte vectorial. Los resultados se equiparan con el estado del arte para opiniones en español.

Palabras clave: máquinas de soporte vectorial, orientación semántica, bigramas morfosintácticos, función kernel.

1. Introducción

La enorme cantidad de comentarios de libre acceso en la Web, para productos y servicios, ha permitido que esas opiniones sean un recurso valioso para tomar decisiones. Las personas interesadas en dichos productos pueden realizar compras basadas en información de otros clientes. Por su volumen, resulta difícil su manejo para un análisis manual, ya que requiere mucho tiempo escoger y comparar opiniones sobre productos objetivo del interés de los compradores. Adicionalmente, dentro de las opiniones, puede haber oraciones que expresen tanto juicios positivos como negativos sobre características del producto, lo que hace más complicada su utilización.

La minería de opiniones intenta clasificar opiniones de forma automática, en función de lo que expresa cada autor. Esta área combina técnicas del Procesamiento de lenguaje natural y de la Minería de textos, y comprende actualmente una gran cantidad de tareas, unas con mayor desarrollo que otras. Por ejemplo, [1] consideraron: identificación de opiniones, polaridad del sentimiento, resumen de la orientación de la opinión; [2] consideró: análisis de sentimiento en oraciones de comparación, detección de SPAM, detección de opiniones que no evalúan, detección de opiniones engañosas.

Una de las principales tareas consiste en determinar la polaridad de las opiniones del documento completo, de oraciones, o de características consideradas. El resultado es clasificar las opiniones en positivas o negativas. Los principales enfoques para resolver este problema corresponden a la clasificación más amplia del aprendizaje mediante

computadora: métodos supervisados y no supervisados. Aunque este problema ha sido estudiado extensamente, la clasificación por polaridad sigue siendo un reto para el procesamiento de lenguaje natural.

En este trabajo aplicamos métodos supervisados a una colección de opiniones en español para lavadoras. Tratándose de productos tan básicos de la vida actual, clasificar estas opiniones es un reto porque los autores de estas opiniones utilizan lenguaje coloquial principalmente, incluyen pasajes anecdóticos y no ponen atención en introducir los signos de puntuación necesarios. Primero seguimos la idea de [3] que considera bigramas morfosintácticos como características para un método no supervisado. Aquí utilizamos bigramas morfosintácticos como características de métodos supervisados, principalmente Máquinas de soporte vectorial (SVM).

El artículo se organiza como sigue: en la segunda sección se presentan trabajos relacionados, enseguida se presenta una descripción de los materiales empleados y del conocimiento lingüístico considerado. En la cuarta sección se describe la herramienta utilizada para aplicar métodos supervisados y los parámetros que se consideraron. En la quinta sección se describen las condiciones de los experimentos y en la sexta sección presentamos los resultados correspondientes. Finalmente presentamos nuestras conclusiones y planteamos las líneas de trabajo futuro.

2. Trabajos relacionados

Desde que la Minería de opiniones se ha asumido como un reto en el Procesamiento de lenguaje natural, una de las tareas en la que más esfuerzos se ha aplicado es la clasificación de la polaridad de un texto, básicamente si es positiva o negativa. Como lo establecen [1], gran parte del trabajo sobre la clasificación de polaridad se ha llevado a cabo en el contexto de opiniones, pero la entrada a un clasificador de polaridad no es necesariamente siempre una opinión estrictamente.

En ese contexto, las opiniones positivas y negativas son a menudo evaluativas (*gustar*, *disgustar*) pero hay otros problemas en su interpretación, y aunque no proporcionan una definición clara de lo que debe considerarse problemas en la clasificación de polaridad, consideran que: (a) en la determinación de la polaridad de los textos de opiniones, donde los autores expresan explícitamente su sentimiento a través de frases como “*este portátil es grande*”, la información objetiva como “*larga duración de la batería*” se utiliza a menudo para ayudar a determinar el sentimiento completo; (b) la tarea de determinar si un pedazo de información objetiva es buena o mala todavía no es lo mismo que clasificarla en una de las varias clases; y (c) la distinción entre información subjetiva y objetiva puede ser sutil.

Desde el inicio se han considerado modificadores, como los adjetivos. El trabajo de [4] para predecir la orientación semántica de adjetivos considera un algoritmo de agrupamiento que separa los adjetivos en grupos de diferentes orientaciones, basado en la correlación entre características lingüísticas y la orientación semántica. Por ejemplo, adjetivos unidos con la conjunción “y” corresponden a adjetivos de la misma orientación como *justo y legítimo*, en oposición a *tranquilo pero flojo*.

Los principales enfoques que se han propuesto desde entonces para resolver la clasificación de la polaridad corresponden a la clasificación de aprendizaje automático: métodos no supervisados y métodos supervisados. Entre los primeros, un trabajo importante es el de [3], donde se determinó la orientación semántica de bigramas del texto de opiniones en inglés. Esta orientación se usó para calcular la orientación de oraciones y del texto completo de una colección de 410 opiniones de *Epinions*. En el segundo enfoque, el trabajo de [5] analiza los resultados de tres métodos: Naive Bayes, Entropía máxima y SVM en la clasificación de polaridad de opiniones de cine y concluyen que no funcionan tan bien como en categorización de textos por tema. Los autores incluyen distintas características lingüísticas de acuerdo al método; en general consideran: palabras solas, secuencias de dos palabras, categorías gramaticales, posición de la palabra (en las opiniones de películas, en particular, pueden empezar con una declaración general de sentimiento).

En cuanto a trabajos desarrollados para clasificar la orientación semántica de opiniones en español, el trabajo de [6] utilizó el corpus MuchoCine [7], de críticas de cine, para un estudio experimental de combinación de métodos supervisados y no supervisados en un corpus paralelo inglés-español. Tradujeron al inglés el corpus en español y combinaron los dos sistemas obtenidos al aplicar métodos supervisados a cada uno de los corpus con la aplicación de SentiWordNet¹ al corpus en inglés. En [8] los autores proponen un método no supervisado basado en el análisis sintáctico de dependencias para determinar la orientación semántica de textos, donde asignan un valor de orientación semántica a las siguientes construcciones sintácticas: negación, oraciones adversativas subordinadas e intensificadores, además de incluir otras características de las palabras mismas y del texto en general. En el trabajo de [7] compilaron un corpus de casi 4 mil opiniones de cine, y aplicaron el método no supervisado de [3] a una colección de 400 opiniones, 200 positivas y 200 negativas.

3. Materiales empleados y conocimiento lingüístico considerado

3.1. Corpus de opiniones

Para este trabajo usamos la colección de opiniones de [9]. La colección fue compilada automáticamente del sitio *ciao.es* y consta de 2800 opiniones de lavadoras. El tamaño promedio por archivo en lexemas es de 345. El número total de lexemas de la colección es de 854,280. La colección total fue anotada con información de lema y categorías gramaticales (*part of speech* en Inglés) utilizando FreeLing [10], una biblioteca de código abierto.

De la colección total de opiniones en español, extrajimos un subconjunto significativo de opiniones diferentes: 2598 opiniones. No eliminamos las opiniones que claramente son anuncios de empresas de mantenimiento (SPAM) ya que tanto este tipo de textos como las opiniones pagadas por fabricantes aparecen en cualquier colección de opiniones de productos.

¹ SentiWordNet es un recurso léxico que asigna a cada uno de los sentidos de las palabras de WordNet tres valores de orientación (positiva, negativa y objetiva)

Utilizamos esta colección para entrenar un estimador cuyo objetivo es la predicción. Es decir, determinar qué tan bueno es un producto en base a la orientación semántica de las opiniones, y el puntaje de los usuarios que corresponden a: malo (una estrella), regular (dos estrellas), bueno (tres estrellas), muy bueno (cuatro estrellas) o excelente (5 estrellas).

Las opiniones tienen tanto errores gramaticales como ortográficos y de puntuación, pero debido a la diversidad de errores decidimos no aplicar métodos de corrección. Freeling es capaz de dar una categoría gramatical correcta aún con errores de ortografía aunque marca como nombre propio sustantivos comunes que no están precedidos por punto y comienzan con mayúscula.

Las características de esta colección en cuanto a número de opiniones por puntaje se presentan en la Tabla 1. Como se observa y como podría esperarse de opiniones de aparatos electrodomésticos cuyo uso es tan generalizado por su gran utilidad, las opiniones favorables son mayores en una proporción de 6:1 entre opiniones positivas y negativas.

Tabla 1. Corpus de reseñas de artículos comerciales.

Opiniones	Detalles	
	Clase	Numero de instancias
Excelente	1190	5
Muy bueno	838	4
Bueno	239	3
Regular	127	2
Malo	204	1

En el área de aprendizaje automático se ha abordado el problema del desequilibrio de clases en la cantidad de ejemplos de entrenamiento para cada una de ellas y las soluciones que se han dado se han clasificado por [11] de la siguiente manera:

- Modificación del algoritmo. Este enfoque está orientado a la adaptación de métodos de aprendizaje para que sean más sensibles a los problemas de desequilibrio de clases, por ejemplo [12]
- Asignación de pesos distintos a los ejemplos de entrenamiento, introduciendo diferentes costos a ejemplos positivos y negativos. Este enfoque considera costos más altos para la clasificación errónea de la clase de mayoría respecto a la clase minoritaria durante el entrenamiento de clasificadores, por ejemplo: [13]
- Muestreo heterogéneo de datos, incluye bajo-muestreo, sobre-muestreo y métodos híbridos. El bajo-muestreo elimina instancias de clases mayoritarias mientras que el sobre-muestreo crea nuevas instancias de la clase minoritaria. Los métodos híbridos combinan los dos métodos anteriores, por ejemplo: [14]

La herramienta que empleamos en este trabajo permite hacer sobre-muestreo, sin embargo lo hemos dejado para un trabajo futuro, considerando que la proporción de desequilibrio de nuestra colección no es un caso que requiera métodos específicos de balanceo. Nos basamos en los resultados obtenidos por [15], donde para una colección

con similar proporción de desequilibrio no obtiene mejoras con diferentes métodos de balanceo.

3.2. Bigramas afirmativos

La clasificación del sentimiento o de polaridad en opiniones se ha trabajado en diferentes dominios, por ejemplo: reseñas de películas, comentarios de productos, para retroalimentar clientes, como opiniones de hoteles y servicios turísticos [1]. Mucho del trabajo se ha realizado con algoritmos de aprendizaje de máquina. El trabajo realizado con métodos no supervisados, es decir, métodos que no cuentan con ejemplos previamente anotados con la clasificación que se quiere aprender se ha basado en el conteo de términos positivos y negativos, determinando automáticamente si el término es positivo o negativo.

[3] determinó la orientación semántica de una opinión mediante un algoritmo que primero extrae bigramas, es decir, secuencias de dos palabras donde una de ellas es un modificador. Enseguida toma cada bigrama para realizar una búsqueda en la Web empleando el operador NEAR de AltaVista para encontrar cuántos documentos tienen ese bigrama cerca de un término positivo (*excellent*) y de un término negativo (*poor*). El puntaje para los dos conjuntos se realiza mediante la medida de información mutua puntual (Pointwise Mutual Information, PMI). La diferencia de puntuación para los dos conjuntos se utiliza para determinar la puntuación para la orientación semántica (SO-PMI), que da como resultado el grado en que cada (bigrama, término) es positivo o negativo.

El puntaje PMI de dos palabras w_1 y w_2 se obtiene mediante la probabilidad de que las dos palabras aparezcan juntas dividida por las probabilidades de cada palabra en forma individual:

$$PMI(w_1, w_2) = \log[P(w_1, w_2)/(P(w_1)P(w_2))] \quad (1)$$

Considerando el número de hits (resultados obtenidos de la Web para la búsqueda), el cálculo de la orientación semántica fue realizado de la siguiente manera:

$$SO(\text{frase}) = \log_2 \frac{\text{hits}(\text{frase NEAR } \textit{excellent}) \text{ hits}(\textit{poor})}{\text{hits}(\text{frase NEAR } \textit{poor}) \text{ hits}(\textit{excellent})} \quad (2)$$

La orientación semántica de bigramas fue utilizada para determinar orientación semántica de oraciones y opiniones completas. Turney tomó 410 comentarios de *epinions.com*. Los resultados oscilaron entre el 84 % para las revisiones de automóviles y el más bajo de 66 % para las críticas de películas.

Si los bigramas morfosintácticos son una buena característica para métodos no supervisados, suponemos que para métodos supervisados incluso podrían ser mejores. Así que en este trabajo, consideramos los siguientes bigramas morfosintácticos como características para el entrenamiento del método supervisado:

1. Sustantivo - adjetivo
2. Verbo - adverbio
3. Adverbio - adjetivo
4. Adjetivo - adverbio

A diferencia de [3] no aplicamos restricción sobre la palabra que sigue a cada bigrama. En el trabajo de [7] aplicaron el método no supervisado de [3], y por la desproporción obtenida de sus resultados (35,5% para las negativas y 91,5% para las positivas) buscaron un umbral de forma supervisada, utilizando el 80% del corpus de 400 críticas para entrenamiento y el 20% restante para evaluación. En este trabajo empleamos las SVM.

Estos bigramas que llamamos morfosintácticos, no corresponden a compuestos obtenidos de un analizador sintáctico. Escribimos un programa que obtiene de la colección completa de opiniones de lavadoras todas las secuencias de dos palabras cuyas categorías gramaticales cumplen los patrones antes indicados. En el caso sustantivo–adjetivo el programa que extrae estos bigramas comprueba la concordancia en género y número. Para todos los bigramas se extraen no las palabras sino los lemas, esto permite agrupar diversas formas en una sola característica. Por ejemplo: *prenda vaquera* y *prendas vaqueras*, *lavadora nueva* y *lavadoras nuevas*, se agrupan en un solo bigrama para cada par.

Consideramos los bigramas adverbio–adjetivo y adjetivo–adverbio porque aun cuando en español es más común la forma adverbio–adjetivo encontramos que la forma inversa está presente en algunas opiniones de esta colección.

4. Herramienta de clasificación

Una ventaja de las SVM es la gran variedad de funciones kernel que pueden usarse para la clasificación. Es decir, se puede generalizar aun en presencia de muchas características con un amplio margen, usando funciones de nuestro espacio de hipótesis [16]. Lo anterior infiere el uso de heurísticas como *Grid search* para la optimización de un conjunto de hiper–parámetros para un estimador en el espacio de un algoritmo de aprendizaje.

Para una tarea de clasificación es necesario separar los datos entre un conjunto de entrenamiento y un conjunto de prueba, en nuestro caso separamos el corpus de opiniones en 70 % para entrenamiento y 30 % para prueba. Cada ejemplo o instancia se asocia a una clase, categoría o etiqueta, es decir el 70 % de los datos de entrenamiento fueron etiquetados con la clase correspondiente (i.e. *muy buena*) y el 30 % de los datos no tiene etiqueta.

4.1. Preprocesamiento de datos

Una de las ventajas de usar un lenguaje de programación de propósito general como *Python* y herramientas de aprendizaje automático como *scikit learn* es la gran cantidad de bibliotecas robustas para implementar distintos métodos y la manipulación de datos. Para obtener las estructuras lingüísticas definidas, se generaron distintas expresiones regulares para empatar esos patrones. Posteriormente, el programa desarrollado generó archivos del tipo *csv* con los bigramas propuestos en la sección anterior. Decidimos utilizar este formato porque es ampliamente usado por la comunidad de aprendizaje

automático; por ejemplo: la gran mayoría de los datos de UCI *machine learning repository* se encuentran en formato *csv*².

Una vez que se generaron los archivos *csv*, se procesaron con *pandas*, una biblioteca para el manejo y análisis de datos de forma eficiente. *Pandas* ofrece estructuras de datos y operaciones eficientes para tablas relacionales, grandes cantidades de datos y conjuntos de datos etiquetados [17]. En este trabajo utilizamos algunas funciones de esta biblioteca para leer el corpus y presentárselo al clasificador. Posteriormente estos datos fueron separados bajo un esquema de validación cruzada con la finalidad de prevenir el sobreajuste del modelo.

4.2. Sistema

Para resolver este problema de clasificación de opiniones en idioma Español decidimos usar un algoritmo supervisado. La clasificación se hizo mediante SVM para el caso multiclase, dadas sus fuertes bases teóricas y porque las SVM son algoritmos de aprendizaje que tienen la capacidad de aprender independientemente de la dimensionalidad del espacio de características. El objetivo de las SVM es producir un modelo basado en los datos de entrenamiento que prediga las clases o categorías de un conjunto nuevo de instancias, mediante la generación de un hiperplano en un espacio de n -dimensión.

Por [16] se sabe que las SVM funcionan bien para clasificar texto, básicamente porque cuando se clasifica texto se trabaja con espacios de dimensión alta, con pocas características irrelevantes, con representaciones vectoriales dispersas y porque la mayor parte de problemas de clasificación de texto son linealmente separables. Todos estos argumentos son fundamentos teóricos que nos permiten saber que es posible clasificar texto con esta clase de algoritmos supervisados.

5. Experimentos

En los siguientes experimentos comparamos el rendimiento de las máquinas de soporte vectorial para el caso multiclase usando distintas funciones kernel, luego comparamos contra un clasificador aleatorio que genera predicciones respetando la distribución de los datos de la clase del conjunto de entrenamiento. El entrenamiento de la máquina de soporte vectorial fue realizado empleando la herramienta *scikit-learn* una biblioteca de propósito general que implementa una gran variedad de algoritmos de aprendizaje automático y que al igual que otras bibliotecas incorpora o envuelve a la biblioteca de C++ LibSVM [18].

5.1. El truco del kernel

En aprendizaje automático los métodos kernel son una clase de algoritmos para analizar patrones, frecuentemente usados en máquinas de soporte vectorial. Las SVM

² <http://archive.ics.uci.edu/ml/>

necesitan funciones kernel debido a que en espacios de baja dimensión presentan dificultades para clasificar datos. Por ejemplo en R^2 la recta que separa dos conjuntos de vectores se vuelve más compleja si agregamos clases y vectores, por lo tanto necesitamos mapear estos datos a espacios de mayor dimensión donde podamos crear de forma más sencilla hiperplanos que separen instancias o vectores de diferentes clases.

Las funciones kernel se usan en muchos algoritmos de aprendizaje para dar una conexión entre linealidad y no linealidad, es decir para que los algoritmos clasifiquen n conjuntos de datos o vectores de forma eficiente. Si la cota entre C clases de vectores es muy cercana donde la cota es la distancia entre los vectores, el algoritmo tardaría mucho tiempo en converger a una solución y aunque converja a alguna, esta no sería la mejor.

En lugar de tomar la costosa y difícil ruta de clasificar los datos en dimensiones bajas generando una curva que separe las clases, las funciones kernel mandan o mapean los datos representados como vectores a espacios de dimensión mayor con la intención de que esos datos sean separados de forma más sencilla, más aun, estas funciones podrían mapear los datos a espacios de dimensión infinita, por lo que no hay restricciones en la dimensión a la cual se mapean los datos. Esto también es conocido como el truco del kernel.

Si los datos vienen estructurados adecuadamente, el algoritmo fácilmente genera una separación o hiperplano entre las distintas clases de datos. El truco del kernel es una herramienta útil que puede aplicarse a algún algoritmo de aprendizaje automático que dependa de un espacio con producto interno entre n vectores. Cuando mapeamos texto o datos a espacios de dimensión mayor o infinita, tenemos que usar algoritmos que usen el producto interno de vectores en espacios de dimensión alta y que sean capaces de generar un hiperplano. Para calcular el producto interno de vectores en espacios de dimensión infinita podemos usar una función kernel que calcule el producto interno directamente usando vectores que viven en espacios de baja dimensión.

Con el fin de probar estos algoritmos para el análisis de patrones usamos los kernels más usados en máquinas de soporte vectorial y evaluamos su comportamiento.

- Kernel de función de base radial. Este kernel mapea los datos a un espacio de dimensión infinita

$$k(x, y) = \exp(-\gamma \|x - y\|^2) \quad (3)$$

- Kernel polinomial: Esta función kernel es útil para problemas en los cuales los datos están normalizados:

$$k(x, y) = (\alpha x^T y + c)^d \quad (4)$$

- Kernel lineal: Es la función kernel más simple. Es calculada por el producto interno $x^T y$, yí más una c constante:

$$k(x, y) = x^T y + c \quad (5)$$

5.2. Evaluación

El sistema se configuró con tres funciones kernel distintas (i.e. lineal, polinomial, función de base radial) y se fueron agregando las características propuestas en la sección 3 para posteriormente evaluar la calidad de la clasificación mediante las siguientes métricas:

Exactitud: En clasificación multiclase, esta métrica calcula el subconjunto de la precisión del conjunto de etiquetas predichas para una muestra que exactamente corresponden al conjunto de etiquetas del conjunto de entrenamiento.

F1-score: Puede ser interpretado como un promedio balanceado entre la precisión y el recall, una F1-score alcanza su mejor valor en 1 y su peor valor en 0. La contribución relativa de precisión y recall al F1-score son iguales.

Score: Se refiere a la media de la precisión, dados los datos y etiquetas de prueba.

Recall: Es la capacidad que tiene un estimador de encontrar todas las muestras positivas. El recall es la relación $t_p/(t_p + f_n)$ donde t_p es el número de verdaderos positivos y f_n es el número de falsos negativos.

Precisión: Intuitivamente podemos decir que es la capacidad que tiene un estimador de no etiquetar como positiva una muestra que es negativa. Precisión es la relación $t_p/(t_p + f_p)$ donde f_p es el número de falsos positivos.

Pérdida de Hamming: Es la fracción promedio de etiquetas incorrectas en porcentaje. Nótese que la pérdida Hamming es una función de pérdida y que el valor perfecto es cero.

Similaridad de Jaccard: El índice de Jaccard o el coeficiente de similaridad de Jaccard, es el tamaño de la intersección dividida por el tamaño de la unión de dos conjuntos de etiquetas, esto es útil para comparar el conjunto de etiquetas predichas para una muestra correspondiente a un conjunto de etiquetas en los datos de entrenamiento.

F-Beta Score: Esta métrica es la media armónica balanceada entre la precisión y el recall, alcanzando su óptimo valor en 1 y su peor valor en 0. El parámetro beta determina el peso de la precisión en el valor de la calificación.

5.3. Grid search

Las SVM particularmente son sensibles al conjunto de hiperparámetros con las que son entrenadas. Los hiper-parámetros o la configuración de los distintos estimadores no siempre pueden encontrarse directamente por un estimador, por eso es necesario hacer una Grid Search para encontrar la mejor configuración, es decir la que aporte la mejor exactitud, F1-score, score, recall, precisión, pérdida de Hamming, similaridad de Jaccard, F-Beta score. Una Grid Search consiste en:

- Un estimador
- Un espacio de parámetros

- Un método para buscar o muestrear candidatos
- Un esquema de validación cruzada

Una Grid search es una búsqueda exhaustiva a través de un subconjunto del espacio de hiper-parámetros de un algoritmo de aprendizaje. Un algoritmo de Grid search debe ser guiado por una métrica de rendimiento, típicamente medido por validación cruzada en el subconjunto de entrenamiento [19]. Al ser una búsqueda exhaustiva la grid search sufre de *The curse of dimensionality*³ pero generalmente es *embarrassingly parallel*⁴ por lo que puede ocurrir una explosión combinatoria que retrase la búsqueda del mejor conjunto de hiperparámetros en el espacio de un algoritmo de aprendizaje automático.

5.4. Evaluando el rendimiento base

Dado que muchas tareas de aprendizaje automático y minería de datos tratan de incrementar la tasa de éxito de resolución de un problema (i.e. tareas de clasificación). Evaluar la tasa base de éxito puede aportar un valor mínimo que otro estimador debe superar. Para comparar el resultado usamos un clasificador que usa estrategias muy simples (aleatorio y siempre predice la etiqueta más frecuente en el conjunto de entrenamiento). Este clasificador nos da una medida del rendimiento base del sistema (i.e. la tasa de éxito que uno debería esperar alcanzar aun cuando simplemente este adivinando).

Supongamos que queremos determinar si una opinión tiene o tiene alguna propiedad. Si hemos analizado un gran número de esas opiniones y hemos encontrado que el 90 % contiene la propiedad objetivo, entonces adivinar que cada futura instancia de la opinión objetivo la posea nos da un 90 % de probabilidad de adivinar correctamente.

Esto es equivalente a usar la estrategia de clasificación más frecuente que implementa la herramienta con la que se hizo el entrenamiento. Se obtuvieron los siguientes resultados con el sistema base: exactitud: 0.330767436436, F1 score: 0.33034110814, score: 0.321957945536, recall: 0.330767436436, precisión: 0.329920827829, perdidade Hamming: 0.669232563564, similaridad de Jaccard: 0.330767436436, F-Beta score: 0.200236094347. Los resultados de todos los experimentos se muestran en la Tabla 2, donde se observa que mejoran sobradamente los resultados del sistema base.

6. Resultados

A continuación presentamos los resultados de clasificar las opiniones en idioma

³ En análisis numérico, muestreo, combinatoria, aprendizaje automático, minería de datos y bases de datos se refiere al fenómeno que surge al analizar y organizar datos en espacios de dimensiones muy altas

⁴ Esto significa que se requiere poco o ningún esfuerzo para separar el problema en una serie de tareas paralelas

español mediante distintas configuraciones de máquinas de soporte vectorial con las características propuestas representadas con TF-idf.

En la figura 1 mostramos el rendimiento para un kernel lineal y un polinomial, debido a que el rendimiento del kernel RBF fue muy parecido al polinomial. En el análisis de resultados nos referimos a los valores obtenidos para el kernel lineal ya que fueron mejores aproximadamente en una relación de 2:1 a los otros dos.

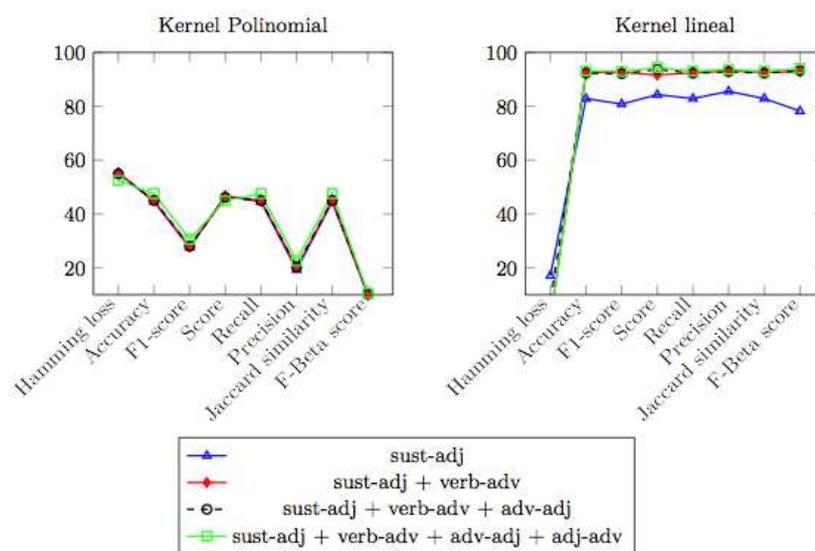


Fig. 1. Rendimiento del sistema con las distintas configuraciones y características propuestas.

Separamos la aplicación de los bigramas como características de entrenamiento en los siguientes conjuntos:

1. Sustantivo-adjetivo
2. Sustantivo-adjetivo y Verbo-adverbio
3. Sustantivo-adjetivo, Verbo-adverbio y Adverbio-adjetivo
4. Sustantivo-adjetivo, Verbo-adverbio, Adverbio-adjetivo y Adjetivo-adverbio

El conjunto 1 considera únicamente el bigrama Sustantivo-adjetivo que cumple una relación sintáctica, ya que como lo indicamos previamente se verificaron concordancias de género y número. La razón para iniciar con este bigrama es que expresa atributos de sustantivos que corresponderían a atributos de características del producto. La evaluación obtenida muestra resultados importantes: exactitud de 82.86 y F-Beta score de 78.22.

El conjunto 2 adiciona el bigrama Verbo-adverbio mediante el cual se puede expresar el modo en que se realiza la acción descrita por el verbo. La adición de este bigrama mejora la clasificación en casi un 10 %, la exactitud pasa de 82.86 a 92.65 y F-Beta score de 78.22 a 92.85.

El conjunto 3 adiciona a los anteriores bigramas el correspondiente a Adverbio-adjetivo. Este experimento muestra que este bigrama no es de utilidad. La exactitud

retrocede de 92.65 a 92.30 y F-Beta score pasa de 92.85 a 93.23 Consideramos este bigrama porque cuando el adverbio se une con un adjetivo, su función semántica es cualificadora o cuantificadora. Sin embargo, en estas opiniones su aportación a la clasificación es menor en relación a otros bigramas.

Tabla 2. Resultados de los experimentos.

Característica	Máquina de soporte vectorial			
	Métricas	RBF	Polinomial	Lineal
Bigrama Sustantivo- adjetivo	Exactitud:	46.96	44.63	82.86
	F1 score:	30.02	27.55	80.83
	Score:	45.22	46.37	84.31
	Recall:	46.96	44.37	82.86
	Precisión:	22.06	19.22	85.56
	Perdida de Hamming:	53.03	55.36	0.171
	Similaridad de Jaccard:	46.96	44.63	82.86
	F-Beta score:	10.50	10.03	78.22
	Sustantivo - adjetivo Verbo - adverbio	Exactitud:	47.08	44.87
F1 score:		30.14	27.79	92.44
Score:		45.86	46.26	91.60
Recall:		47.86	44.87	92.44
Precisión:		22.17	20.13	93.08
Perdida de Hamming:		52.91	55.12	0.073
Similaridad de Jaccard:		47.08	44.87	92.65
F-Beta score:		10.53	10.08	92.85
Sustantivo - adjetivo Verbo - adverbio Adverbio - adjetivo		Exactitud:	45.10	46.03
	F1 score:	28.04	29.02	92.08
	Score:	46.14	45.68	93.73
	Recall:	45.10	46.03	92.30
	Precisión:	20.34	21.19	92.86
	Perdida de Hamming:	54.89	53.96	0.076
	Similaridad de Jaccard:	45.10	46.03	92.30
	F-Beta score:	10.13	10.32	93.23
	Sustantivo - adjetivo Verbo - adverbio Adverbio -	Exactitud:	44.63	47.55
F1 score:		27.55	30.64	92.99
Score:		46.63	44.94	94.36
Recall:		44.63	47.55	93.12
Precisión:		19.92	22.61	93.12

adjetivo				
Adjetivo - adverbio	Perdida de Hamming:	55.36	52.44	0.068
	Similaridad de Jaccard:	44.63	47.55	93.12
	F-Beta score:	10.03	10.62	94.07

El conjunto 4 adiciona el bigrama Adjetivo-adverbio, el cuál no corresponde a un orden muy común en el español. Sin embargo, algunas secuencias como *perfecto desde luego, mejor claro, super bien*, mejoran los resultados de la clasificación. La exactitud pasa de 92.30 a 93.12 y F-Beta score de 93.23 a 94.07

Aunque la comparación con otros trabajos no puede ser directa por las colecciones empleadas, las características de entrenamiento, y los parámetros de los métodos supervisados, a continuación indicamos algunos resultados de clasificación de polaridad de opiniones en español.

En [8] utilizan una colección de 25 opiniones favorables y 25 opiniones desfavorables para lavadoras y con un método no supervisado obtienen 0.88 de *Accuracy* para opiniones positivas y 0.76 para opiniones negativas. En [6] aplican el método SVM a la colección de opiniones de cine de [7] y obtienen *Precision* de 0.8771, *Recall* de 0.8763, F1 de 0.8767 y *Accuracy* de 0.8766. Estos resultados muestran que los aquí obtenidos con bigramas se equiparan con el estado del arte de opiniones en español.

7. Conclusiones

En este artículo describimos experimentos realizados sobre un corpus de opiniones en español para obtener la orientación semántica de cada opinión. Analizamos el impacto de los bigramas morfosintácticos que definimos y extrajimos de la colección sobre el funcionamiento de un método supervisado para la clasificación de dicha orientación semántica. Los resultados muestran el impacto de cada bigrama, y especialmente el aporte menor del bigrama adverbio-adjetivo. En un trabajo futuro incluiremos bigramas de connotación negativa y haremos experimentos para balancear la colección.

Exploramos la utilidad de la biblioteca Scikit-learn de Python ya que este paquete es totalmente abierto y reutilizable. Debido a que Python se ha convertido en un lenguaje de programación ampliamente utilizado en Procesamiento de lenguaje natural consideramos importante explorar su funcionamiento para aprendizaje supervisado. Las funcionalidades implementadas nos permitieron hacer diversas evaluaciones y asignación de valores a parámetros del método SVM.

Referencias

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, vol. 2 (1-2), pp.1–135 (2008)

2. Liu, B.: Sentiment analysis and subjectivity. *Handbook of natural language processing*, pp. 627–666 (2010)
3. Turney, P. D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics (ACL)*, pp. 417–424 (2002)
4. Hatzivassiloglou, V., Kathleen, R. M.: Predicting the Semantic Orientation of Adjectives. In *EACL '97 Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pp. 174–181 (1997)
5. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of EMNLP*, pp. 79–86 (2002)
6. Martín-Valdivia, M. T., Martínez-Camara, E., Perea-Ortega, J.M., Ureña López, L.A.: Sentiment polarity detection in spanish reviews combining supervised and unsupervised approaches. *Expert Systems with Applications Expert Systems with Applications*, vol. 40 (10), pp. 3934–3942 (2013)
7. Cruz Mata, F., Troyano Jiménez, J. A., Enríquez de Salamanca, R., Ortega Rodríguez, F.J.: Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. *Procesamiento del lenguaje natural*, vol. 41, pp. 73–80 (2008)
8. Vilares, D., Alonso, M. A., Gómez-Rodríguez, C.: A syntactic approach for opinion mining on Spanish reviews. *Natural Language Engineering*, vol. 1(1), pp. 1–26 (2013)
9. Galicia-Haro, S. N., Gelbukh, A.: Extraction of Semantic Relations from Opinion Reviews in Spanish. In *Human-Inspired Computing and Its Applications. Lecture Notes in Computer Science*, vol. 8856, pp. 175–190. Springer (2014)
10. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey (2012)
11. López, V., Fernández, A., García, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, vol. 250, pp.113–141 (2013)
12. Sun, Y., Kamel, M. S., Wong, A., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, vol. 40 (12), pp. 3358–3378 (2007)
13. Pazzani, M., Merz, C., Murphy, P., Kamal, A., Hume, T., Brunk, C.: Reducing misclassification costs. In *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 217–225 (1994)
14. Tang, Y., Yan-Qing, Z., Nitesh, V. C., Krasser, S.: Svms modeling for highly imbalanced classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39(1), pp. 281–288 (2009)
15. Rehan, A., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In *Machine Learning: ECML 2004*, pp. 39–50 (2004)
16. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. Springer (1998)
17. McKinney, W.: Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pp. 51–56 (2010)
18. Chih-Chung, C., Chih-Jen, L.: LibSVM: a Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2(3), p. 27 (2011)
19. Chih-Wei, H., Chih-Chung, C., Chih-Jen, L.: A practical guide to support vector classification. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>