

# Detección del lenguaje figurativo e ironía en textos cortos

Patricia Espinoza, Darnes Vilariño, Mireya Tovar, and Beatríz Beltrán

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
Avenida San Claudio, 14 Sur, Ciudad Universitaria.  
Puebla México.  
[patricia.efong@gmail.mx](mailto:patricia.efong@gmail.mx)  
[darnes,mtovar,bbeltran@cs.buap.mx](mailto:darnes,mtovar,bbeltran@cs.buap.mx)  
<http://nlp.cs.buap.mx>

**Resumen** En la presente investigación se propone un modelo para resolver la Tarea 11 de la Competencia Semeval 2015. El modelo propuesto utiliza características léxicas extraídas de los textos, así como la polaridad de las palabras obtenidas utilizando diferentes herramientas. El modelo fue validado con 1 corpus de Twitter y se compara el desempeño de dos de los algoritmos más usados para clasificación: Naïve Bayes Multinomial y Máquinas de soporte vectorial.

**Palabras clave:** Análisis de sentimientos, twitter, polaridad

## 1. Introducción

Analizar la polaridad de los textos es una tarea importante para varias aplicaciones de la vida real. Existe un gran número de tareas que se ven beneficiadas por el desarrollo de métodos computacionales que permitan detectar la intención con que ha sido escrito un cierto mensaje, es decir, si el mensaje es positivo, negativo, neutro u objetivo. Una de las aplicaciones directas de esto se aprecia cuando se desea construir la reputación de un político, de una celebridad o de un hombre de negocios que se puede hacer a partir del análisis de la polaridad de los mensajes en Internet, sin embargo esta tarea ha evolucionado del análisis de mensajes donde se usa un lenguaje literal, sin doble sentido, hasta detectar ese lenguaje figurativo o ironía en el texto, ya que si una opinión es clasificada como positiva existe la posibilidad de que sea irónica. En particular en este trabajo se presenta una comparación de los resultados obtenidos al desarrollar un modelo para la Tarea 11 del semeval 2015 [1]. La cual consiste en crear un sistema de análisis de sentimientos que nos ayude a identificar el grado de lenguaje figurativo e ironía implementada dentro de un mensaje. Dicho modelo fue probado con dos de los clasificadores más usados actualmente implementados en Weka<sup>1</sup>: Naïve Bayes Multinomial<sup>2</sup> y Máquinas de soporte vectorial (SMO)<sup>3</sup>.

<sup>1</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

<sup>2</sup> <http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayesMultinomial.html>

<sup>3</sup> <http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html>

La estructura del artículo es la siguiente. En la sección 2 se presentan los trabajos desarrollados en la literatura con respecto a la identificación de emociones en textos cortos. La sección 3 presenta la descripción de las características y el modelo de clasificación usando los experimentos desarrollados. La discusión acerca de los resultados obtenidos se presenta en la sección 4. Finalmente la conclusión del presente trabajo de investigación se realiza en la sección 5.

## 2. Trabajo Relacionado

En general la mayoría de los trabajos reportados hasta el momento para darle solución a esta tarea se han encaminado a la extracción de características estadísticas, el uso de diccionarios de expansión y la generación de modelos supervisados de clasificación. A continuación se comentan los trabajos más relevantes reportados en el Semeval 2013. En el trabajo desarrollado por [4] se extraen como características los unigramas y bigramas de palabras, se toma en cuenta una colección de hashtags positivos y negativos. Para las entradas se generaron parejas de unigrama–unigrama, de unigrama–bigrama y de bigrama–bigrama, se eliminó el carácter “@”. Con las características extraídas se desarrolló un modelo de clasificación utilizando la Máquina de Soporte Vectorial(SVM), a través del paquete LibSVM, utilizando un kernel lineal. Este modelo incluye además como características el número de emoticones, los signos de puntuación, el número de palabras con mayúsculas, número de signos de negación y los nombres de usuarios en twitter. Con las características seleccionadas reportan una precisión que supera el 60 %.

En la investigación realizada en [3] se propone el desarrollo de un motor de detección<sup>4</sup> que está diseñado específicamente para detectar el sentimiento positivo, negativo o neutral en tweets, para lograr esto se realiza la detección de rasgos semánticos del tweet como emoticones y onomatopeyas, la evaluación de los hashtags y la descomposición del tweet en una lista de n–gramas, se realiza una comparación de cada n-grama con los términos léxicos. Se plantea que dado que los emoticones y onomatopeyas tienen fuertes indicios de sentimiento, pero también tienen una gran variedad ortográfica, se tiene una lista con las exclamaciones más comunes y se utilizan expresiones regulares para capturar la variedad de formas que pueden asumir. Para la evaluación de los hashtags se aplica una serie de heurísticas para dividirlos (en caso de que el hashtag sea compuesto) . El tweet es descompuesto en una lista de unigramas, bigramas, trigramas y tetragramas, se recorren todos los n-gramas del tweet y se realizan comprobaciones de su presencia en diccionarios, finalmente para el análisis de sentimientos se utilizaron cuatro glosarios: tono positivo, tono negativo, fuerza del sentimiento y negaciones; los cuales fueron creados manualmente. En este trabajo se reporta una precisión mayor al 60 %.

Otro trabajo que es importante destacar es el desarrollado por [7]. Para detectar la polaridad de un tweet, se establece que esto es posible si se aplican los

---

<sup>4</sup> Disponible en [www.umigon.com](http://www.umigon.com)

siguientes pasos de normalización: todas las palabras se convierten a minúscula, se sustituyen @ y # por las notaciones de [usuario], y de [tag] respectivamente, los emoticones se clasifican en positivos y negativos, se remueven caracteres innecesarios, en el caso de palabras que contienen repeticiones de caracteres se reduce la longitud, sólo teniendo en cuenta una secuencia de tres caracteres, con el fin de unificar estas repeticiones. Después de la normalización de los mensajes se determinó la polaridad de cada palabra utilizando el diccionario de sentimientos SentiWordNet, se considera una palabra como positiva si el valor positivo relacionado es mayor a 0.3; como negativa, si el valor negativo relacionado es mayor a 0.2 y como neutral si el valor relacionado es mayor a 0.8. Una vez calculada la polaridad se contemplan tres características para cada tweet que son: el número de palabras positivas, negativas y objetivas respectivamente; se verifica si a una palabra positiva le precede una negación, si es así la polaridad se invierte; se utilizó un diccionario de siglas y para cada sigla se utilizó una polaridad. Esta propuesta supera el 50 % de precisión.

En el trabajo desarrollado por [2], se propone que cada mensaje debe ser preprocesado considerando que los mensajes deben ser tokenizados utilizando una expresión regular, que coincide con direcciones URL, secuencias de caracteres alfanuméricos o secuencias de caracteres sin espacio, de esta forma las secuencias de puntuación como los emoticones se conservan, se separan las palabras en caso de que falte un espacio; los hashtags pueden ser procesados por separado o como n-gramas. Para la clasificación de los mensajes en las clases positiva, negativa y neutra, se utilizaron tres modelos lineales y fueron entrenados uno contra todos, en tiempo de predicción se eligió el marcador con puntuación más alta. Los modelos lineales fueron capacitados mediante el uso del método de optimización gradiente estocástico. El término estocástico se refiere al hecho de que los pesos del modelo se actualizan para cada modelo de entrenamiento.

A pesar de que se han desarrollado diversas investigaciones, los resultados obtenidos hasta el momento en esta propuesta no superan una precisión superior al 60 %.

En [6] proponen un modelo para detectar la ironía en textos obtenidos de Twitter, el cual es organizado alrededor de 4 tipos de características conceptuales:

1. Firmas: engloba elementos tipográficos como signos de puntuación, emoticones y palabras en mayúscula.
2. Lo inesperado: aquí se estudian contradicciones que pudieran ocurrir en la oración, midiendo la similitud semántica de los conceptos del texto.
3. Estilo: está representado obteniendo las frecuencias de estos 3 tipos de secuencias, los n-grams, los skip-grams y los polarity-grams.
4. Escenario emocional: se intenta capturar información emocional de las palabras a través de un diccionario.

Ellos reportan resultados de 73 % para un corpus balanceado y 62 % para un corpus no balanceado.

Otro trabajo relacionado a la detección de ironía es [5], en donde se estudian opiniones de productos de un sitio de ventas en línea. Se usan como caracte-

terísticas los n-grams de caracteres para encontrar términos frecuentes, POS n-grams cuyo objetivo es encontrar secuencias de patrones morfosintácticos, ya que la ironía usa aspectos divertidos para crear su efecto, entonces ellos buscan características humorísticas que describan a un documento. También se calcula la relación entre elementos positivos y negativos en el texto y algunas otras emociones. Los resultados reportados por ellos son del 87% utilizando el algoritmo de Naïve Bayes y del 89.03% con Máquinas de soporte vectorial.

### 3. Descripción del enfoque propuesto

En el enfoque propuesto se emplea un modelo supervisado basado en máquinas de aprendizaje, para el cual se construye un modelo de clasificación usando la metodología que se muestra en la figura 1.

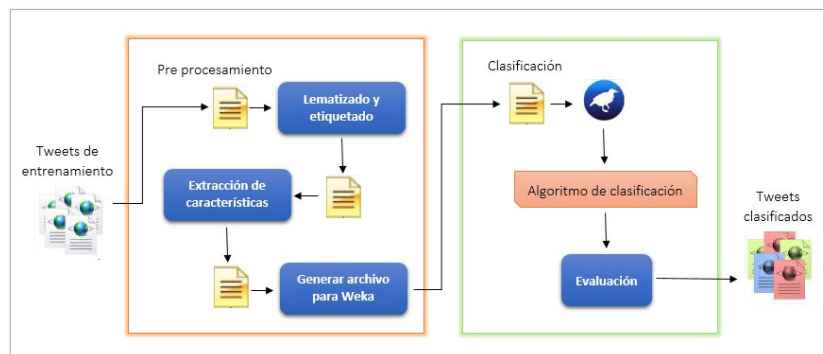


Fig. 1. Metodología empleada.

Como se puede observar, la metodología propuesta consta de dos fases las cuales se explican con mas detalle a continuación:

#### 3.1. FASE 1 - Lematizado y etiquetado

Para realizar este proceso se utilizó la herramienta de Clips llamada `pattern.en`<sup>5</sup>, la cual cuenta con un método llamado "parse" el cual identifica las oraciones, las palabras y la categoría a la que pertenece una palabra dentro de una oración. Este parser utiliza un lexicon de 100,000 palabras conocidas y su correspondiente categoría. Un ejemplo de la implementación de esta función se muestra en la figura 1.

<sup>5</sup> <http://www.clips.ua.ac.be/pages/pattern-en>

```
>>> from pattern.en import parse
>>> print parse('I eat pizza with a fork.')

I/PRP/B-NP/O eat/VBD/B-VP/O pizza/NN/B-NP/O with/IN/B-PP/B-PNP a/DT/B-NP/I-PNP
fork/NN/I-NP/I-PNP ././O/O
```

Fig. 2. Ejemplo de uso de la función parse de pattern.

### 3.2. FASE 1 - Extracción de características

Las características generales obtenidas del corpus de Twitter proporcionado para esta tarea son las siguientes:

1. Conteo de características léxicas.
2. Conteo de categorías gramaticales.
3. Polaridad y objetividad de una oración (Clips).
4. Polaridad de las palabras (SentiWordNet).
5. Emociones de las palabras (NRC Emotion Lexicon).
6. Bolsa de palabras.

La descripción de cómo se calcula cada característica, con el objetivo de construir un vector representativo de cada mensaje, se explica a continuación.

Las características léxicas están compuestas por las siguientes 15 características:

1. Número de slangs.
2. Número de contracciones.
3. Número de prefijos.
4. Número de signos.
5. Conteo de las 100 palabras más utilizadas.
6. Número de palabras mal escritas.
7. Longitud de la oración.
8. Número de números.
9. Número de palabras que empiezan con mayúscula.
10. Número de palabras escritas en mayúscula.
11. Longitud de la palabra mas larga.
12. Número de palabras de longitud 1,2,10 y 15.

Para estos conteos se desarrollaron diferentes recursos léxicos como son: un diccionario de slangs, un diccionario de signos, diccionario de contracciones, diccionario de prefijos, diccionario de palabras más frecuente y un diccionario que nos permite detectar si la palabra ha sido mal escrita . Las categorías gramaticales generan un vector de 39 características ya que esas son las que nos proporciona Clips.

Para detectar el grado de ironía y el lenguaje figurativo se parte de la hipótesis descrita en[6] que plantea que en un texto irónico hay mas elementos positivos que negativos.

Para intentar probar esta hipótesis se utilizan diferentes recursos léxicos para realizar un conteo de cargas positivas y negativas en la oración. Uno de ellos es la función de análisis de sentimientos de Clips con la cual podemos obtener la polaridad y la objetividad del texto, basado en los adjetivos que este contenga. Se sumaron todos los valores de polaridad y objetividad de cada palabra de la oración y se agregaron estas dos como características al vector. Otro recurso léxico que se utilizó y que está disponible para su descarga de forma gratuita, es SentiWordNet<sup>6</sup>, del cual se obtuvieron los valores positivos y negativos de cada palabra del texto, ambos valores se agregan al vector como 2 características separadas. Para calcular el valor positivo y negativo de cada oración, se sigue el mismo procedimiento que con el anterior. También se utilizó el recurso léxico NRC Emotion Lexicon<sup>7</sup>, este describe las palabras con valores binarios a partir de 10 emociones (enojo, anticipación, asco, miedo, alegría, negativo, positivo, tristeza, sorpresa, confianza) , como por ejemplo la representación de la palabra "happy" se puede ver en la tabla 1.

**Tabla 1.** Ejemplo de la representación de las palabras en NRC

Palabra	Emoción	Valor
happy	anger	0
happy	anticipation	1
happy	disgust	0
happy	fear	0
happy	joy	1
happy	negative	0
happy	positive	1
happy	sadness	0
happy	surprise	0
happy	trust	1

Se agregan 10 características que suman el valor de las 10 emociones de cada tweet. Por último se agrega el mensaje como una bolsa de palabras, haciendo un total de 69 características.

### 3.3. FASE 1 - Generar archivo para weka

Con este vector finalmente se construye un archivo el cual sirve para crear un modelo de clasificación , donde el atributo clasificador varía desde -5.0 hasta 5.0, obteniéndose un total de 11 posibles clases -5, -4, -3, -2, -1, 0, 1, 2 , 3, 4, 5, siendo -5 un tweet totalmente irónico o de doble sentido y 5 un tweet con lenguaje literal.

### 3.4. FASE 2 - Clasificación

Utilizando el archivo creado en la fase anterior, se utilizan los algoritmos de clasificación Naïve Bayes Multinomial y Máquinas de soporte vectorial (SMO) implementados en Weka.

<sup>6</sup> <http://sentiwordnet.isti.cnr.it/>

<sup>7</sup> <http://www.saifmohammad.com/WebPages/ResearchInterests.html>

## 4. Resultados Experimentales

El corpus de entrenamiento se compone de textos breves (mensajes), una descripción completa del conjunto de datos de entrenamiento se encuentra en el artículo de descripción de la tarea 11[1].

En la tabla 2 se pueden apreciar los resultados obtenidos, se realizaron diferentes experimentos para analizar el impacto de cada característica sobre el conjunto de entrenamiento. Se puede observar el desempeño de ambos algoritmos. Se aplicó validación cruzada con 10 pliegues.

El mejor comportamiento se presentó al utilizar el clasificador máquina de soporte vectorial, con una precisión del 43.12%. Se considera que este resultado se obtuvo como consecuencia de que el corpus está totalmente desbalanceado y el tamaño de los tweets difiere mucho, lo que provoca que ciertas características no puedan ser obtenidas. Las investigaciones que han obtenido mejores resultados reportan que los corpus evaluados con su metodología tienen características muy particulares y sobre todo como son opiniones de usuarios de amazon el tamaño de cada opinión es mucho más grande.

**Tabla 2.** Resultados obtenidos

<b>System</b>	<b>NaïveBayes</b>	<b>Multinomial SMO</b>
Léxicas	36.95	37.33
Léxicas+Gramaticales	38.57	38.28
Léxicas+Gramaticales+Clips	38.51	38.82
Léxicas+Gramaticales+Clips+SentiWordNet	38.37	38.44
Léxicas+Gramaticales+Clips+SentiWordNet+Bolsa	41	<b>43.12</b>
Léxicas+Gramaticales+Clips+SentiWordNet+Bolsa+NRC	41.6	42.71

## 5. Conclusiones

Se ha presentado una aproximación para detectar el grado de lenguaje figurativo o ironía de un mensaje usando básicamente tres tipos de características: polaridad y emoción de las palabras, conteos de diferentes elementos y palabras del mensaje como tal. Los resultados obtenidos muestran que estas características no son suficientes para detectar correctamente la clase de un mensaje con alta precisión. Se considera que el desbalanceo de los datos de entrenamiento, así como el número de clases que se tienen en total, influyó significativamente en los resultados que se han obtenido. Sin embargo, se está analizando los conjuntos de datos para lograr encontrar nuevas características, que permitan desarrollar un mejor modelo de clasificación .

## Referencias

1. John Barnden, Antonio Reyes, Ekaterina Shutova, Paolo Rosso, and Tony Veale. Semeval–2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval–2015)*, Dublin, Ireland, 2015.

2. Tobias Gunther and Lenz Furrer. Gu–mlt–lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.
3. Clement Levallois. Umigon: Sentiment analysis for tweets based on lexicons and heuristics. In *Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 520–524, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.
4. Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc–canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.
5. Antonio Reyes and Paolo Rosso. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision Support Systems*, 53(4), november 2012.
6. Antonio Reyes, Paolo Rosso, and Tony Veale. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1), march 2013.
7. Richard Farkas Viktor Hangya, Gabor Berend. Szte–nlp: Sentiment detection on twitter messages. Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.