# Modelos de grafos para la detección de datos de texto no estructurados como el sarcasmo

## *Graph Model for Detection of text unstructured data such as Sarcasm*

ARTICLE HISTORY

Received 11 October 2020
Accepted 02 November 2020

**Axel Rodríguez-García**
Facultad de Ingeniería de Sistemas Computacionales
Universidad Tecnológica de Panamá
Ciudad de Panamá, Panamá
axel.rodriguez2@utp.ac.pa

**Armando Jipsion**
Facultad de Ingeniería de Sistemas Computacionales
Universidad Tecnológica de Panamá
Ciudad de Panamá, Panamá
armando.jipsion@utp.ac.pa

# Modelos de grafos para la detección de datos de texto no estructurados como el sarcasmo

## *Graph Model for Detection of text unstructured data such as Sarcasm*

**Axel Rodríguez-García**
Facultad de Ingeniería de Sistemas Computacionales
Universidad Tecnológica de Panamá
Ciudad de Panamá, Panamá
axel.rodriguez2@utp.ac.pa

**Armando Jipsion**
Facultad de Ingeniería de Sistemas Computacionales
Universidad Tecnológica de Panamá
Ciudad de Panamá, Panamá
armando.jipsion@utp.ac.pa

***Abstract*—** Sarcasm is frequently characterized as verbal incongruity to communicate scorn. It is a nuanced type of language with which people express something contrary to what is suggested. Perhaps the greatest test in building frameworks to consequently recognize unstructured information, for example, mockery, is the absence of huge, commented on informational indexes. We propose a diagram-based procedure in building conservative language models for sarcasm recognition. This strategy is likewise intended to utilize little information, it could help in different regions like disdain discourse, counterfeit news, and so forth. This charting strategy permits specialists to explore different parts of NLP without obtaining a huge dataset. These days, it still remains a challenge to unmistakably distinguish human slants and feelings by utilizing AI. Associations can use a superior philosophy to settle on proactive choices in basic circumstances. A definite investigation of our examination would hoist the current content mining applications and may help understand better the effect of mockery from the customers and partners communicated in a web-based media climate. We exhibit that straightforward classifiers worked from the model can recognize mockery very well, which they sum up 5 % better than those of the cutting edge.

***Keywords*—** *Unstructured Data, NLP, sarcasm, modelo de grafo.*

***Resumen*—** El sarcasmo se define a menudo como una ironía verbal para expresar desprecio, un lenguaje matizado con el que los individuos expresan lo contrario de lo que está implícito. Uno de los mayores retos en la construcción de sistemas para detectar los datos no estructurados como el sarcasmo, es la falta de grandes conjuntos de datos anotados. Proponemos un método basado en grafos para la construcción de modelos de lenguaje compacto para la detección del sarcasmo. Este método está diseñado para usar pocos datos, y podría ayudar a detectar fake news, hate speech, etc. Permite además a los investigadores analizar otros aspectos del NLP sin tener que obtener un conjunto de datos gigante. Hoy en día, sigue siendo un desafío identificar claramente los sentimientos y emociones humanos mediante el uso de Inteligencia Artificial. Una exploración detallada de nuestra investigación elevaría las aplicaciones actuales de minería de textos y podría ayudar a comprender mejor el impacto del sarcasmo de los clientes y las partes interesadas, expresado en un entorno de redes sociales. Demostramos que los clasificadores simples construidos a partir del modelo pueden detectar bastante bien el sarcasmo, que generalizan un 5% mejor que los del estado del arte.

***Palabras clave*—** *Datos no estructurados, lenguaje de procesamiento natural (NLP), sarcasmo, graph model.*

## I. INTRODUCCIÓN

¿Qué es el sarcasmo? Es una ironía verbal que expresa el desprecio hacia una persona o la ridiculiza. Su naturaleza figurativa dificulta el análisis de sentimientos. Es también una forma de lenguaje matizada en la que los individuos afirman lo contrario de lo que se implica. En el lenguaje hablado, el sarcasmo se puede identificar por el tono del interlocutor, pero solo en el texto, detectar el sarcasmo se vuelve muy difícil. En términos generales, el sarcasmo implica un sentimiento negativo, pero a menudo muestra un sentimiento positivo en en cuanto a



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



mejorar la atención del cliente [1]. Por ejemplo: cuando un cliente descontento, que no pudo obtener un servicio satisfactorio de Amazon, hizo un comentario en Twitter publicando: "Buen trabajo@ AmazonHelp". O cuando una persona le dice a su amigo que: si este es un hotel de 5 estrellas, entonces "Sí, y yo soy la Reina de Inglaterra ". Tanto "Grantrabajo@..." como "Sí, y soy..." son sarcasmos en el contexto.

La detección automática del sarcasmo es una tarea para predecir el sarcasmo en el texto. Esto se ha vuelto cada vez más importante para mejorar el rendimiento de los sistemas de análisis de los sentimientos. Aparte del desafío de disuadir; o minar el verdadero sentimiento en una frase sarcástica [1], otro desafío importante para la detección automática del sarcasmo es la falta de grandes y fiables conjuntos de datos anotados [2]. La mayoría de los conjuntos de datos que existen fueron creados a partir de textos en inglés, lo que deja a otros idiomas con pocos recursos para abordar esta difícil tarea.

En este artículo, proponemos una metodología basada en grafos para construir modelos compactos de lenguaje para la detección del sarcasmo. Utilizando la supervisión a distancia mediante el uso del hashtag #sarcasmo, se recolectó un pequeño conjunto de datos de unos pocos tweets de mil arenas. Los textos fueron luego convertidos a forma de grafo, y con el uso de técnicas de análisis de grafos, los patrones fueron descubiertos automáticamente. Luego se extrajeron las representaciones vectoriales de los patrones; utilizando un enfoque para aprender las representaciones latentes de los vértices en una red.

El modelo de lenguaje compacto resultante se utilizó luego para construir un clasificador detectando los patrones en los textos de un conjunto de entrenamiento, obteniendo sus incrustaciones (embeddings) y representando cada texto de muestra como el promedio de todas las incrustaciones. Comparamos nuestro método con las líneas de base, que se basaban en las últimas técnicas de construcción de modelos de lenguaje y en los enfoques más avanzados para la detección del sarcasmo. Demostramos experimentalmente que con pocos datos, nuestro enfoque de construcción de modelos, puede generar clasificadores aceptables que generalizan mejor a los métodos más avanzados. Este enfoque podría extenderse a otros idiomas utilizando la traducción apropiada para "sarcasmo".
Las principales contribuciones de nuestro trabajo son:

1) Un enfoque independiente del idioma para construir un modelo para la detección del sarcasmo.

2) El enfoque propuesto puede construir modelos efectivos con pocos datos utilizando la supervisión a distancia.

3) Debido al tamaño compacto del modelo, puede construirse sin necesidad de una gran potencia de cálculo.

4) El modelo de lenguaje puede ser usado para obtener características para clasificadores simples basados en la Regresión Logística o Máquinas Vectoriales de Apoyo.

5) El modelo lingüístico es lo suficientemente expresivo como para ayudar a los clasificadores construidos sobre él, y para generalizar mejor al de los enfoques más complejos.

El resto de esta invetigación está estructurada de la siguiente manera: La sección 2 puntualiza el marcco teórico. En la sección 3 se especifica la metodología. Los experimentos serán presentados previamente en la sección 4. Por último, las conclusiones y el trabajo futuro figuran en la sección 5.

## II. MARCO TEÓRICO

La tarea de clasificación es la metodología más común de la detección automática del sarcasmo. Varios estudios han abordado la cuestión de la detección del sarcasmo en textos cortos (por ejemplo, tweets). Al tratar esos textos, algunos investigadores anotaron manualmente los tweets como obras sarcásticas o no sarcásticas [1]. Se basaron en la supervisión a distancia para crear conjuntos de datos. El trabajo en [3] utiliza un conjunto de datos creados mediante la obtención de tweets que contienen hashtags, como #sarcasmo, #sarcasmo, #no. Otros estudios que utilizan la supervisión a distancia incluyen [4],[5],[6],[7],[8],[9],[10],[11],[12].

Otros estudios que utilizan este enfoque incluyen el trabajo en [3], con la diferencia de que sólo mantienen los tweets que contienen el hashtag en cuestión al final. No todos los textos que contienen sarcasmo son tweets o mensajes cortos de medios sociales. Los foros de debate y las reseñas de productos, que suelen ser más extensos, también pueden contener muchos casos de sarcasmo que vale la pena detectar. En la obra presentada en [11]. En [11] se utiliza un conjunto de datos de mensajes de foros de debate con múltiples etiquetas que incluyen el sarcasmo. De acuerdo con la investigación en [13] se utiliza un conjunto de datos de reseñas

de películas, libros y artículos de noticias marcados con sarcasmo y otras etiquetas de sentimiento. Otros trabajos de investigación que utilizan reseñas etiquetadas con sarcasmo incluyen [14], [15], [16], [17],[18].

Por último, dado que el sarcasmo aparece a menudo en conversaciones reales, también se han realizado investigaciones sobre transcripciones y diálogos. Estos pueden ser divididos en transcripciones de centros de llamadas [17] y frases extraídas de las conversiones de ciertos programas de televisión [18], [19]. La mayoría de los trabajos sobre detección automática de sarcasmo se han basado únicamente en el texto mismo. Más recientemente, los estudios han experimentado con información adicional para proporcionar el texto de la convención. Esto incluye el contexto específico del autor. Considerando la historia del autor, [20], [21] incorpora en el conjunto de datos tweets pasados como contexto. Otra forma de contexto cuando se trata de conversaciones incluye frases que ocurren antes de la frase a clasificar de una conversación [21], [22],[23],[24],[25] o comentarios relacionados con un post en foros de discusión [21]. Otro enfoque interesante para obtener el contexto es la idea de que ciertos temas evocan más sarcasmos que otros. Con eso en mente, los estudios presentados en [25], [26], intentan crear modelos de temas para ayudar a predecir el sarcasmo. Estudios más nuevos y avanzados han llevado el contexto a un nivel más profundo al incorporar metadatos como seguidores, ubicación, edad de la cuenta, número de mensajes, gustos, etc., de redes sociales como [15]. Se han utilizado diferentes tipos de características y algoritmos de aprendizaje para abordar la detección automática del sarcasmo. La mayoría de los estudios se basan en modelos de lenguaje de bolsa de palabras, aunque otros han hecho uso de patrones con coincidencia parcial o total.

La mayor parte de los trabajos efectuados en este campo se basan en diferentes formas de algoritmos tradicionales de aprendizaje de máquinas como Support Vector Machines (SVM) o Logistic Regression (Regresión Logística) en [1], [3], [15], [17], [23], [27], [28], [29]. Estudios más recientes han comenzado a utilizar cada vez más los enfoques de aprendizaje profundo para abordar esta difícil tarea [21],[30],[31],[32].

## III. METODOLOGÍA

### *A. Vision General*

Proponemos un enfoque novedoso para construir modelos de lenguaje para la detección del sarcasmo convirtiendo el conjunto de datos en un grafo y extrayendo de él representaciones de nodos (embeddings). Las representaciones resultantes pueden utilizarse para entrenar a un clasificador. Aunque este documento se centra en la detección de datos no estructurados como el sarcasmo en Twitter, el mismo enfoque podría ajustarse fácilmente para otras tareas.

Para generar el modelo, es necesario seguir los siguientes pasos. En primer lugar, el conjunto de datos se transforma en un grafo considerando las palabras (y en algún caso especial, secuencias cortas de palabras llamadas patrones) como nodos, y se crean conexiones entre los nodos (arcos o aristas) para las palabras que aparecen una al lado de la otra (o dentro de una distancia determinada) en los textos originales. La fig.1 y fig. 2 muestran ejemplos de este proceso de sustracción de grafos, en donde el resultado final es un grafo subjetivo después de eliminar los bordes.

A continuación, se analizan los grafos iniciando recorridos aleatorios desde cada nodo y generando secuencias aleatorias de los nodos que pueden verse como frases sintéticas. Se utiliza una técnica de generación de modelos lingüísticos basados en el texto para extraer incrustaciones (embedding) de nodos, tratando los patrones como una oración regular. El modelo de lenguaje resultante puede entonces ser usado con los clasificadores tradicionales de aprendizaje automático (Machine Learning).

***B. Construcción del grafo:*** El primer paso para construir modelos es generar versiones de grafos de los textos. En este estudio, se investigan tres formas diferentes de construir dichos grafos.

***1) Pathways Graphs (Grafos de Caminos):*** El primer enfoque para construir un grafo se basa en el enfoque descrito en [33]. Los pasos de preprocesamiento incluyen:

1. Normalizar el texto poniendo en minúsculas el conjunto de datos

2.Eliminar las palabras de parada (opcional)

3.Rama (opcional)

Tradicionalmente, la etapa de preprocesamiento, o de limpieza, incluye la normalización de las palabras mediante la corrección de errores ortográficos o la reducción de fichas con letras repetidas (por ejemplo wooooow!!!!!). No incluimos esto como parte del preprocesamiento; estas fichas especiales pueden ayudar a la detección del sarcasmo.



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



Una vez que el conjunto de datos ha sido preprocesado, el grafo se construye escaneando los textos, creando un nodo para cada palabra única, y bordes entre las palabras de co-ocurrencias dentro de un espacio de n palabras. El hueco es pre-especificado antes de la construcción y se pueden utilizar diferentes tamaños del hueco.

Un frase en el idioma inglés como: "today is very sunny and hot" generaría los siguientes aristas si la brecha se fijara en 1: today→ is, today→ very, is →ivery, is→ sunny, etc.

Aunque el ejemplo anterior muestra un→
para indicar el borde, esto es sólo para la representación, ya que el grafo real no está dirigido y no está ponderado.

***Definición 3.1*** Grafo de Palabra -Word Graph :Definimos el grafo de palabras de la siguiente manera:
donde V denota una agrupación de nodos de

$$G = (V, E, W) \qquad (1)$$

palabras, E representa la agrupación de las relaciones entre dos nodos de palabras, y W representa el conjunto de pesos de sus bordes (edge).

Definición 3.2. Peso de la arista - Edge Weight: Para cada arista $e_{vi,vj}$ ϵ E conección de palabras $v_i$ y $v_j$, el peso del borde $w_{vi,vj}$ es calculado como:
donde freq ($e_{vi,vj}$}) denota como la frecuencia

$$W_{vi,vj} = \frac{freq(e_{vi,vj})}{\max(freq(E))} \qquad (2)$$

de palabras vi and vj co-ocurrencia y freq(E) denota el conjunto de frecuencias entre todos los pares de palabras.

En el caso de los Pathways Graphs (Gráfos de Caminos), no hay pesos para las aristas.

***2) Minusnet Graphs:*** El segundo tipo de grafo se basa en partes de los métodos presentados en[34]. Su método se usó para crear clasificadores de emociones y estaba destinado a ser un enfoque independiente del idioma. Dado que el sarcasmo es un tipo de emoción, y dado que el paso particular descrito en ese documento podría funcionar en general para cualquier tarea, decidimos adaptarlo para construir nuestro segundo tipo de grafo.

La idea general detrás de los Grafos Minusnet es que el grafo resultante del conjunto de datos original contendrá muchos nodos y bordes inútiles. Podemos pensar en estos objetos sin importancia como palabras y co-ocurrencias que son concurrentes para cualquier lenguaje dado y no transmiten información útil para la tarea en cuestión. Por lo tanto, eliminar estos objetos del grafo podría hacer el modelo más compacto y aumentar su poder de predicción.
Para crear un Grafo Minusnet, se necesitan dos conjuntos de datos. El primero es el conjunto de datos relacionados con la tarea en cuestión, en el caso de este trabajo es un conjunto de datos con textos sarcásticos. El segundo conjunto de datos debe ser más general (neutral en el caso de la detección de emociones), y no contener instancias del primer conjunto de datos. Ya que el sarcasmo es un tipo de emoción, usamos un conjunto de datos neutrales (titulares de noticias) como en [34].

Para construir el grafo, ambos conjuntos de datos se convierten primero en un grafo ponderado utilizando el enfoque del Grafo de Caminos descrito en la definición del grafo de palabra. La principal diferencia esta vez es que los bordes tendrán un peso igual a su frecuencia dividido por la máxima frecuencia de borde en el grafo como en la Definición de Peso del arco. Por lo tanto, el borde más frecuente tendrá un peso de 1,0, y cada otro borde tendrá un peso entre 0,0 y 1,0. Después de completar este proceso obtenemos dos grafos Gn=(Vn,En,Wn) y Gs=(Vs,Es,Ws) donde el subíndice n denota neutralidad y s sarcasmo.

A continuación, un enfoque de sustracción de grafos introducido en [34] se aplica para generar el grafo final de Minusnet. La sustracción simplemente encuentra los bordes que coinciden en ambos grafos y resta el peso del borde en el grafo neutro del que está en el grafo de sarcasmo.

***Definición 3.3*** [Graph Subtraction- Substracción del Grafo] Dado un grafo de sarcasmo Gs=(Vs,Es,Ws) y un grafo neutral Gn=(Vn,En,Wn), nuevos pesos Ws se calculan para Gs de la siguiente manera:

$$Ws_{vi,vj} = \begin{cases} Ws_{vi,vj} - Ws_{vi,vj} & if\ vi,vj \in Vs \cap Vn \\ Ws_{vi,vj} & otherwise \end{cases} \qquad (3)$$

Después de actualizar un peso en el grafo de sarcasmo, si el valor resultante está por debajo de un cierto umbral; el arco correspondiente se elimina del grafo. Si, después de eliminar los arcos, algún nodo (palabra); ya no está conectado al grafo, el nodo se elimina. Esto da como resultado un grafo y las figuras 2 y 3 del Minusnet muestran un ejemplo de este proceso.

***3) Patterns Graphs (Grafos de Patrones):*** El tercer y último tipo de grafo es el único en el que los nodos no son fichas de una sola palabra. En su lugar, los nodos son secuencias cortas de fichas de palabras y un comodín. Tabla ref{patterns-examples} muestra algunos ejemplos de patrones.

La idea es encontrar los patrones más comunes que se producen en los datos, y para aumentar el poder de generalización, sustituir algunas palabras por un comodín (.*) que coincida con cualquier palabra. Como por ejemplo en la fig. 3. Esto podría verse como un tipo de n-gram con un poder extra dado por el comodín. Aparte del comodín, lo que hace que los patrones sean diferentes de los n-gramas tradicionales es la forma en que se descubren.

El enfoque de descubrimiento de patrones se introdujo por primera vez en [34] y funciona de la siguiente manera: Un grafo Minusnet se construye a partir de una agrupación *de datos objetivos y una agrupación de datos neutral*. Dos tipos de palabras se extraen del grafo mediante técnicas de análisis de grafos. El primer tipo es *Connector Palabras*, que son frecuentes y también centrales en el grafo. El segundo tipo es *Palabras del tema*, que son menos frecuentes y tienden a agruparse con otras palabras relacionadas con los mismos temas. Utilizando las dos listas se construyen diferentes combinaciones de palabras de longitud de variables (instancias). Se mantienen todas las secuencias de instancias que se producen en el conjunto de datos y se hace un seguimiento de su frecuencia. Las instancias frecuentes tendrán su palabra temática reemplazada por el comodín para convertirse en un patrón. Después de este reemplazo, diferentes instancias se convertirán en el mismo patrón *"I love this"* y *"I hate this"* se convertirán en "I .* this" si "love" y "hate" son ambos *Topic Words (Palabras del tema))*.
Los patrones y su frecuencia se mantienen. Luego, se btiene una lista final de patrones seleccionando aquellos cuya frecuencia está por encima de un umbral determinado. A partir de la lista final de patrones se construye un Grafo de Patrones utilizando una versión modificada de la construcción del Grafo de Patrones. Más detalles para cada uno de los pasos anteriores son los siguientes:

Para encontrar el *Connector-Conector y Topic Words-Palabras* Tópicas, El análisis de los grafos es necesario. *Connector Words (conector de palabras)* son un tipo especial de palabras de parada, sin embargo, no están predefinidas y no se consideran una molestia. En cambio, *Connector Words* son descubiertos a partir del conjunto de datos; son frecuentes y son centrales, por lo que desempeñan un papel importante en el poder de generalización de los patrones resultantes. Para descubrir estas palabras, *betweenness centrality (Análisis de Centralidad)* el análisis se realiza en el grafo. En la teoría de los grafos, betweenness centrality es una medida de la centralidad en un grafo basado en los caminos más cortos. Una palabra con mayor betweenness centrality sería un buen candidato para un componente de patrón general, porque más n-gramas lo contendrán. La razón para usar *Connector Words*, en lugar de palabras de parada es que las palabras de parada son limitadas en número y expresión. *Betweenness centrality* se define como:

***Definición 3.4*** [Betweenness Centrality (Medida de Centralidad)] Por casa palabra en el nodo vu, la medidad de centralidad es calculada como:

$$BCv_u = \sum_{v_u \neq v_i \neq v_j} \frac{\sigma_{vi,vj}(v_u)}{\sigma_{vi,vj}} \qquad (4)$$

where $\sigma_{vi,vj}$ es la cantidad total de caminos más cortos desde el nodo vi al nodo vj y $\sigma_{vi,vj}(v_u)$ es el número de esos caminos que pasan por $v_u$.

***a) Finding Topic Words (Encontrando palabras temáticas) :*** Las palabras temáticas se consideran como aquellas que tienen un alto grado de significado. En un gráfo de palabras, se espera que las palabras temáticas que tienen temas similares tiendan a agruparse. También se cree que si las palabras llevan menos significado, o en nuestro caso están menos relacionadas con la tarea de detectar el sarcasmo, estarán más aisladas. Para medir el grado de agrupación en un grafo, y detectar Topic Words, se calcula el coeficiente de agrupación de cada nodo. Siendo el coeficiente de agrupación una medidad del grado, en que los nodos de un grafo tienden a agruparse. Podemos definir coeficiente de agrupación de acuerdo a la formula # (5). De la siguiente manera:

***Definición 3.5*** [Clustering Coefficient (Coeficiente de la agrupación)] Por cada palabra del modo vu , el coeficiente es calculado como:

$$CCv_u = \frac{2T(Vu)}{nei(Vu)(nei(Vu)-1} \qquad (5)$$

Donde T ($v_u$) es el número de triángulos a través de los vértices $v_u$y nei($v_u$) es el números de vecinos de $v_u$.



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



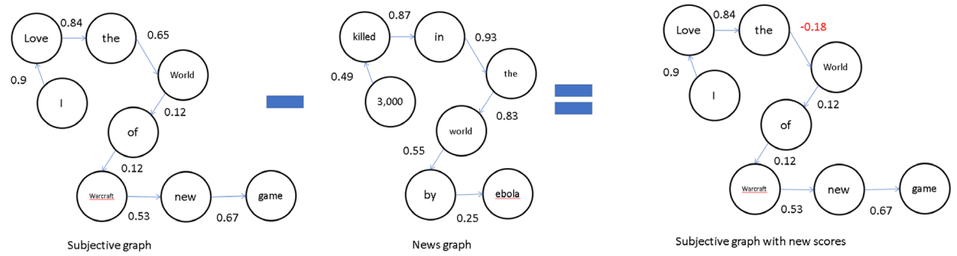


*Fig. 1. Sustracción entre un grafo subjetivo y un grafo de titulares de noticias.*

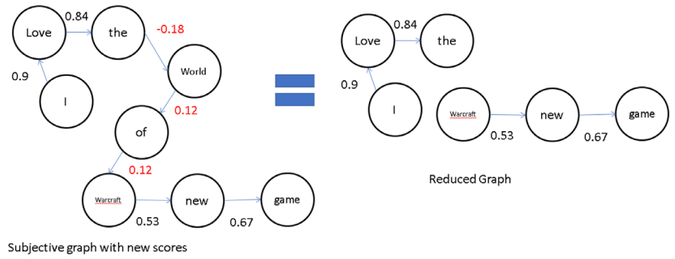


*Fig. 2. Grafo subjetivo final después de eliminar los bordes (y los nodoscorrespondiente) con un peso inferior a un valor de umbral de 0,2.*

| Pattern | Example sentence match |
|---|---|
| was .* 🤣 | That was professional 🤣 |
| 👍 .* . | Well done 👍 Trump. |
| loves .* and | He loves burgers and pizza |

*Fig. 3. Ejemplo de patrones y frases que los contienen. La coincidencia está subrayada en la frase.*

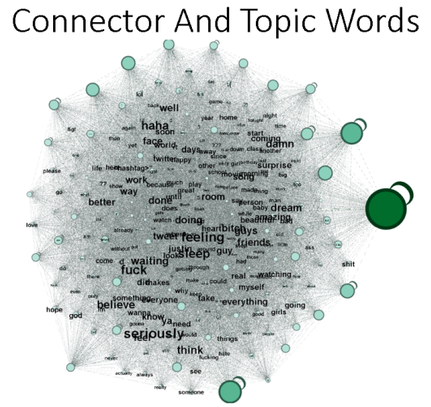


*Fig. 4. Conector y palabras del tema dentro del grafo. El tamaño del nodo se determina por la medida de la centralidad mientras que el tamaño de la etiqueta se determina por el coeficiente de agrupación. Se puede observar como las palabras con etiquetas más grandes tienen más significado.*

La fig. 4 muestra algunos ejemplos del conector y el tema Palabras encontradas en un gráfico Minusnet.

***b) Extracting Instances (Extracción de instancias) :*** Los patrones en el conjunto de datos se descubren de abajo hacia arriba. La idea es encontrar primero el texto de los patrones potenciales, donde por instancias nos referimos a una secuencia de palabras que se corresponderían con tales patrones, y luego deducimos los patrones a partir de las frecuencias de instancias. El primer paso es definir un modelo de instancias, o del acuerdo a la invesigación original de acuerdo a [34] llamado *meta-patterns.* A *meta-pattern "CW-TW-CW"* indica que instancias debe construirse probando todas las combinaciones de Conector de Palabras Connecting Words, seguido de *Topic Words (Palabras Tópicas)*, y por último seguido finalmento por un *Conector de Palabras.*

*Meta-patterns* definir tanto el orden en que aparecen los dos tipos de palabras, como la longitud de la *instancias.* A *meta-pattern* es al menos 2 palabras y debe contener al menos una de cada tipo de palabras. Todas las *instancias* Todos meta-patterns (patrones meta). Para determinadas instancias (se puede pensar en ellos como n-gramas), calculamos su frecuencia en el conjunto de datos. Fig. 5 muestra ejemplos de ambos tipos de palabras y las *instancias* que puede resultar de ellas.

| Topic Words |
|---|
| love |
| hate |
| gift |
| weather |
| ... |

| Connector Words |
|---|
| this |
| got |
| my |
| pain |
| ... |

| Instances |
|---|
| "hate this weather" |
| "lots of pain" |
| "got my gift" |
| "love this drawing" |
| "got my price" |
| "am in pain" |
| "kill this idiot" |
| "finish this task" |

*Fig. 5. Dos tipos de palabras y algunas instancias.*

***c) Generating Patterns (Generación de patrones):*** Desde las instancias con un proceso sencillo. En primer lugar, todas las instancias con la misma longitud y teniendo el mismo *Connector Word (Conector de palabra)* (la misma ficha de superficie en la misma posición) se agrupan. Sus frecuencias se agregan en la frecuencia del grupo. Entonces *pattern (el patrón)* se crea reemplazando los Topic Words *(Palabras Tópicas)* (que dentro de un grupo debe estar siempre en la misma posición para cada instancia) con una ficha especial que representa el comodín (e.g. * or.*).
Después de que cada grupo se ha convertido en un *pattern (patrón)*, no de frecuencias *patterns(patrones)* se filtran. Las Fig. 6 y 7 muestran los dos pasos que acabamos de describir.

| Instances | Count |
|---|---|
| "hate this weather" | 5 |
| "lots of pain" | 4 |
| "got my gift" | 7 |
| "love this drawing" | 2 |
| "got my price" | 1 |
| "am in pain" | 3 |
| "kill this idiot" | 1 |
| "finish this task" | 4 |

| Connector Words |
|---|
| this |
| got |
| my |
| pain |
| ... |

| Groups | Count |
|---|---|
| "Hate this weather", "love this drawing", "kill this idiot", "finish this task" | 12 |
| "lots of pain", "am in pain" | 7 |
| "got my gift", "got my price" | 8 |
| ... | ... |

*Fig. 6. Agrupando las instancias mediante el uso del Connector Word.*

| Pattern | Groups | Freq. |
|---|---|---|
| * this * | "Hate this weather", "love this drawing", "kill this idiot", "finish this task" | 12 |
| * * pain | "lots of pain", "am in pain" | 7 |
| got my * | "got my gift", "got my price" | 8 |
| ... | ... | ... |

| Connector Words |
|---|
| this |
| got |
| my |
| pain |
| ... |

*Fig. 7. Obtener patrones manteniendo las Palabras de Conexión y reemplazando las Palabras Temáticas con un comodín.*

La idea principal es que construyendo un grafo reducido de palabras (Minusnet) que sea altamente representativo de la tarea en cuestión, y seleccionando de él palabras que sean a la vez centrales (comunes y a través de las cuales pase mucho significado) y que estén agrupadas (ellas mismas llevan mucho significado), podemos descubrir combinaciones que luego pueden llegar a ser muy expresivas *patterns (patrones)*. Reemplazando el original Topic Word (conector de palabra) con un comodín que coincida con cada palabra tiene el poder potencial de capturar otras palabras temáticas que no estaban presentes en el conjunto de datos original, pero puede ser importante para la tarea en cuestión. En la fig. 8 podemos apreciar el proceso parcial de construcción del grafo Esto es particularmente importante cuando se trata de pequeños conjuntos de datos. En la fig. 8 se puede apreciar el proceso parcial de construcción del grafo observando de dos a tres gramos consecutivos.

@PoptartLudwig Great selection of commentators on this sports channel!

k = 3

| First 3-gram |
|---|
| @poptartludwig great selection |
| **Pattern [Matched portion of the 3-gram]** |
| .+ great ['@poptartludwig great'] |
| great .+ ['great selection'] |
| .+ great .+ ['@poptartludwig great selection'] |
| **Second 3-gram** |
| great selection of |
| **Patterns [Matched portion of the 3-gram]** |
| .+ .+ of ['great selection of'] |
| .+ of ['selection of'] |
| great .+ of ['great selection of'] |
| great .+ ['great selection'] |
| great .+ .+ ['great selection of'] |
| **Edges between patterns with matches in first 3-gram and second 3-gram** |
| .+ great ----> .+ .+ of |
| .+ great ----> .+ of |
| .+ great ----> great .+ of |
| .+ great ----> great .+ |
| .+ great ----> great .+ .+ |
| great .+ ----> .+ .+ of |
| great .+ ----> .+ of |
| .+ great .+ ----> .+ of |
| ... |

*Fig. 8. Proceso parcial de construcción del grafo observando de dos o tres gramos consecutivos. Por cada 3 gramos, las líneas de abajo muestran el patrón y la porción del 3º gramo que fue emparejado. Después de mirar los dos 3 gramos consecutivos, se crea una conexión entre los patrones que coinciden con el primero y los patrones que coinciden con el segundo.La flecha es solo → para ilustración; ya que el grafo no está dirigido.*

***d) Patterns Graph Construction-Construcción de Grafos de Patrones:*** Una vez que el *patterns (patrón)* se extraen, pueden utilizarse para construir un segundo grafo final que constituirá el modelo de lenguaje (en forma de grafo; hay otro paso más adelante para conseguir el embedding o incrustación). El proceso de construcción de este grafo es una versión modificada del enfoque en la construcción de los Grafos de Caminos 3.2.3.

La principal diferencia viene del hecho de que *patterns o patrones* tienen una longitud



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



variable a diferencia de las palabras sueltas.
Para construir el grafo, todos los *k-grams*, conociendo la longitud de los patrones más largo, se extraen de cada texto en el conjunto de datos y el *k-grams* son inspeccionados en secuencia. Típicamente, patterns tienen una longitud de 2 o 3 pero podrían ser más, dependiendo de cómo meta-patterns se definen. La coincidencia de expresiones regulares puede utilizarse para encontrar los patrones que coinciden con un *k-gramo* dado.
La coincidencia puede ser parcial como algunos patterns pueden ser más corto que *k*. Cada dos patterns que coinciden con dos consecutivos *k-grams* se añadirán al grafo como nodos y se generará un borde correspondiente. Las fig. 7 ilustra este proceso y la fig. 9 muestra algunos patterns o caminos clasificado por su *Clustering Coefficient (Coeficiente de la Argupación) dentro de el Patterns Graph (Grafo de patrones).*

| Pattern | Clustering Coefficient |
|---|---|
| best .+ | 1 |
| good .+ | 1 |
| away their .+ | 1 |
| .+ 😁 .+ | 1 |
| ... .+ | 1 |
| .+ 😂 .+ | 1 |
| .+ would | 1 |
| .+ 🤓 .+ | 1 |
| .+ ! | 1 |
| great .+ | 1 |
| happens ... .+ | 1 |
| .+ 😳 .+ | 1 |
| ever was .+ | 0.964285714 |
| will .+ this | 0.945454545 |
| an .+ in | 0.933333333 |
| in these .+ | 0.928571429 |
| already .+ the | 0.927272727 |
| .+ love the | 0.923809524 |
| at .+ best | 0.916666667 |
| are too .+ | 0.916666667 |
| but a .+ | 0.916666667 |
| he be .+ | 0.916666667 |
| her so .+ | 0.916666667 |
| so .+ much | 0.909090909 |
| a title .+ | 0.904761905 |
| .+ imagine why | 0.904411765 |
| just .+ how | 0.891666667 |
| can't imagine .+ | 0.888888889 |
| last .+ years | 0.884615385 |

*Fig. 9. Algunos patrones clasificados por su coeficiente de agrupación dentro del grafo. Los valores van de 0.0 a 1.0.*

***4) Latent Representations Extraction (Extracción de representaciones latentes)***
En la sección anterior, describimos cómo construir diferentes tipos de grafos de texto para representar nuestros modelos de sarcasmo. En esta sección, describimos el proceso de extracción de incrustaciones (embeddings) para las diferentes características de words (las palabras) o patterns (patrones) de los grafos. El enfoque descrito en [35] se utiliza para obtener las diferentes incrustaciones de nodos del grafo. En Deepwalk: Aprendizaje en línea de las representaciones sociales [35], Las representaciones sociales de los vértices de un grafo se aprenden modelando recorridos cortos aleatorios. Las representaciones sociales son características latentes de los vértices que capturan la similitud del vecindario y la pertenencia a la comunidad.

Hemos seleccionado *DeepWalk* como nuestro enfoque en la obtención de incrustaciones para las fichas de nuestro grafo por las siguientes razones:

- ***Adaptabilidad:*** El lenguaje humano está en constante evolución y los temas de moda cambian todo el tiempo. Lo que es una forma popular de expresar el sarcasmo, o un tema popular de evocación de sarcasmo hoy en día, probablemente será más neutral en un futuro próximo. Debido a la forma en que DeepWalk en [34],crea las incrustaciones, las nuevas fichas y conexiones añadidas al grafo pueden ser fácilmente consideradas la próxima vez que se ejecute el proceso de extracción de incrustaciones.

- ***Low dimensional (baja dimensión):*** Como han afirmado los autores del documento en [34] , cuando los datos etiquetados son escasos, los modelos de baja dimensión se generalizan mejor y aceleran la convergencia y la inferencia. Es bien sabido que los datos etiquetados para la detección del sarcasmo son limitados, por lo que se prefieren los modelos de baja dimensión.

El proceso de generación de las incrustaciones con *DeepWalk* comienza con una impulso corto *random walks* utilizado para extraer información de la red. El ciclo de caminos aleatoria toma nuestro grafo de texto G y, en un bucle, toma cada nodo vi de V como la raíz de un camino aleatoria. Para realizar un recorrido, un vecino del último nodo visitado es muestreado uniformemente hasta que se alcanza la máxima longitud del recorrido. La idea principal detrás de *DeepWalk* es que se considerará la secuencia generada por el recorrido como una frase. Esta frase sintética es entonces alimentada al conocido SkipGram algoritmo [34] para generar los embedding (incrustaciones).
Definición 3.6 [Synthetic Sentences as a Random Walk-(Sentencias sintéticas como un paseo aleatorio)]
*Sentencias sintéticas como un paseo aleatorio*

*Por cada raíz del nodo $v_i$, its $k^{th}$ los caminos aleatorios de longitud n + 1 es definido como: donde $v_{s1}$ es tomado aleatoriamennte desde el la vecindad $v_i$, $v_{s2}$ , es tomada aleatoriamente desde la vecindad de $v_{s1}$y así que.*

$$rw_{i,k} = v_{i,}.v_{s1,}\ v_{s2\ldots\ldots,}\ v_{sn} \qquad (6)$$

***5) The Classifier (El Clasificador)***

El objetivo del método propuesto es construir un modelo de lenguaje compacto para la detección del sarcasmo que pueda construirse a partir de pequeños conjuntos de datos y utilizarse con algoritmos sencillos de aprendizaje automático para obtener buenos resultados. Con un modelo de baja dimensión como el descrito en las secciones anteriores, una de las mejores opciones de algoritmo para construir un clasificador es la Regresión Logística (Logistic Regression). Por lo tanto, usamos el modelo para entrenar un clasificador de Regresión Logística y determinamos los mejores parámetros para dicho clasificador experimentalmente.

Para entrenar al clasificador, los tweets en un conjunto de entrenamiento son pre-procesados y marcados como en el paso de construcción del grafo. Los patrones se identifican de la misma manera. Luego, para cada palabra/patrón de un tweet, se extrae la incrustación correspondiente, si existe, y se añade a una lista de incrustaciones para ese tweet en particular. Si cada elemento del tweet no tiene incrustación, el tweet se descarta. La incrustación media del tweet se construye entonces calculando un vector medio obtenido por cálculo de la media a lo largo de las dimensiones de incrustación de la lista de incrustaciones.

El vector medio se convierte entonces en un vector unitario y se utiliza como la incrustación final del tweet. El mismo proceso se aplica cuando se valida.

***Definición 3.7*** [Final Embedding (Incrustración Final)] Para a tweet T, en el que es compuesto de items *(palabras or caminos)* $w_1, w_2, \ldots w_n$. Un embedding para T es entonces definido como: Where $u_{w1}, u_{w2},\ldots, u_{wn}$ son las incrustraciones (embedding) desde nuestro modelo de

$$u_T = \frac{1}{n}\sum_{i=1}^{n} u_{wi} \qquad (7)$$

construcción para cada item de T.

## IV. EXPERIMENTOS Y RESULTADOS

En este estudio se realizaron diferentes experimentos. El objetivo de la primera serie de experimentos es determinar qué tipo de grafo, Pathways, Minusnet, o Patterns, tiene el mejor rendimiento. Para llevar a cabo el experimento, se construyen diferentes grafos de cada tipo cambiando los parámetros de pre procesamiento, aplicando la derivación, se eliminaron las palabras de parada y el tamaño del espacio entre las palabras, etc.

Después de determinar qué tipo de grafo funciona mejor, el segundo conjunto de experimentos tiene como objetivo comparar nuestro mejor modelo con el estado del arte (SOTA, *state of the art*) basadas en técnicas de última generación, así como métodos SOTA para la detección del sarcasmo. Se comparan tres líneas de base y un método SOTA con nuestro enfoque, con el objetivo de determinar qué enfoque generaliza mejor los datos no vistos.

***A. Comparando los tipos de grafos***

En este estudio adoptamos un modelo de lenguaje compacto, basado en grafos de texto, para la detección del sarcasmo. Proponemos tres formas diferentes de construir los gráficos, a saber: *Pathways Graphs-Grafos de caminos, Minusnet Graphs, y Patterns Graphs-Grafo de patrones.* En los dos primeros grafos los nodos son palabras, mientras que en el último los nodos son secuencias especiales de palabras llamadas *patterns o (patrones)*. En esta subsección describimos los experimentos realizados para determinar qué tipo de grafo funciona mejor cuando se utilizan incrustaciones (embeddings) extraídas de él para entrenar un clasificador.

***1) Datasets (Conjunto de Datos)***

Se utilizan diferentes conjuntos de datos durante estos experimentos. Los conjuntos de datos se describen a continuación:

• Conjunto de generación de grafos: este conjunto de datos consiste en 9,550 tweets recopilados con supervisión a distancia mediante el uso del hashtag #sarcasm. Para evitar recopilar por error tweets no sarcásticos utilizando #sarcasmo como parte del texto, nos aseguramos de que todos los tweets tuvieran el hashtag dentro de las últimas 3 palabras. Todos los tweets se clasificaron y recopilaron desde el 11/21/2019 hasta el 02/27/2020.

• Conjunto de datos neutrales: el conjunto de datos neutral consta de 34,047 titulares de noticias en forma de tweets recopilados de las cuentas de los principales medios de comunicación en inglés. Las cuentas provienen de: *The Wall Street Journal, The New York Times, The Guardian, etc.*



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



• Conjunto de datos de emociones: Este contiene 52,207 tweets recopilados con supervisión a distancia usando más de 200 hashtags relacionados con las emociones. Algunos de los hashtags incluyen *#enfado, #excitación, #gozo, #miedo, etc.*

• Conjunto de datos de sarcasmo: Este contiene 18,953 tweets recopilados con supervisión a distancia utilizando el *#sarcasm* hashtags. Estos tweets son diferentes de los c*onjuntos de generación de grafos and #sarcasm* puede aparecer en cualquier parte del tweet.

***2) Configuración del experimento***

Para realizar los experimentos, el conjunto de datos de generación de grafos se utiliza para generar los grafos y extraer las incrustaciones o embeddings. Los otros tres conjuntos de datos se combinan y luego se dividen en un conjunto de entrenamiento y validación para entrenar y validar un clasificador de Regresión Logística. Un 70 %-30 % de la división es utilizada.

Tabla I describe los hiperparametros que han cambiado para construir los grafos. Por ejemplo, cuando construimos *Pathways Graph (El grafo de camino)*, si el paso 2 es seleccionado con cada juego de parámetros verdaderos, entonces por cada *tweet @ AlDeTocqueville* por que este resulta entonces adecuado para nosotros en los pocos años pasados#sarcasm, AlDeTocqueville podría llegar ha *<usermention>, #sarcasm convertir <hashtag>*, las palabras *que, porque , nosotros, etc.* debería ser borrada, y la palabra resultante debería deribarse. Luego, al generar el gráfico, una arista entre los nodos *past y <hashtag>* existiría; ya que podemos tener un espacio de como máximo 2 elementos.

*Tabla I. Los hiperpara´metros para el proceso de generación de grafos.*

| Parameter | Description | Graph Types |
|---|---|---|
| *stepn* | The gap between consecutive words/patterns that will be linked by an edge | All |
| *nostopwords* | If true, stop-words are removed | All |
| *stem* | If true, words are stemmed | All |
| *noentities* | If true, hashtags are changed to a keyword ¡*hashtag*¿ and user mentions are changed to ¡*usermention*¿ | All |
| *diff th* | Minimum weight to keep an edge as part of the *Minusnet graph* | Minusnet, Patterns |
| *cc th* | Minimum *Clustering Coefficient* to consider a word a *Topic Word* | Patterns |
| *centrality th* | Minimum *Betweenness Centrality* to consider a word a *Connector Word* | Patterns |
| *min freq* | Minimum frequency to retain a *pattern* | Patterns |

*Tabla II. Parametros utilizados por deepwalk.*

| Parameter | Description | Default Value |
|---|---|---|
| *randwalks* | Random walks numberS per root node | 10 |
| *walklen* | random walk length | 40 |
| *winsize* | Window size used for *SkipGram* | 5 |
| *embsize* | Size of the resulting embedding | 64 |

La incrustación de cada nodo se calculó usando *DeepWalk* y Tabla II describe los parámetros utilizados. Para todos los experimentos de esta sección, *embsize* se fijó en 64 y 128 mientras se mantenían los otros valores por defecto. Los clasificadores fueron entonces entrenados para clasificar los tweets en sarcásticos, emocionales, or neutrales utilizando el enfoque descrito en 3.4.

***3) Resultados:*** Varios Grafos de camino se construyeron cambiando los parámetros descritos en la Tabla I y utilizando el primer conjunto de datos descrito en 4.1.2. La Fig. 10 muestra clasificadores basados en los Pathways Graphs clasificadas por su precisión en el conjunto de validación.

La fig. 10 resume el resultado de nuestros experimentos en la misma podemos observar que los 3 primeros clasificadores son los que tienen sufijos*_step3_stem_128, _step2_ stem_128, y _step3_noentities_128 con una precisión de 75 %.* Un rápido vistazo a la figura también muestra que, en general, la eliminación de entidades y un tamaño de incrustación de 128 da mejores resultados. También se puede ver que la eliminación de las palabras de parada perjudica la clasificación.

De manera similar, varios *Grafo Minusnet* se construyeron cambiando los parámetros de la misma de manera descrita anteriormente. *El diff_*th se fijó en 0.0 para todos los grafos, lo que significa que los bordes (y potencialmente las palabras) sólo se eliminaron si su peso después del paso de reducción se volvió negativo. Este valor tiene sentido para los conjuntos de datos pequeños, de lo contrario el grafo resultante sería demasiado pequeño.

Podemos observar en la fig. 11 que los 3 primeros clasificadores son los que tienen sufijos *_ step2_noentities_128, _step1_noentities_128, y _step3_stem_noentities_128 todas con precisión alrededor del 75%.* Precisamente como *Grafos de camino*, La eliminación de entidades y un tamaño de incrustación de 128 dan los mejores resultados. La eliminación parece tener un menor impacto y la eliminación de las palabras de parada no ayuda.

Finalmente, varios grafos se construyeron y probaron clasificadores basados en *Grafos de camino.* Este tipo de grafo se introduce unos pocos parámetros nuevos y requiere más tiempo para su construcción. Por lo tanto, no pudimos probar tantas combinaciones de parámetros como deseábamos. En este estudio probamos con *cc_th ...a* 1,0, 2,0 o 3,0, *centrality_th* siempre se ajusta a 0.0001 y min_

*freq ...en* 1 o 3. La Fig. 12 muestra clasificadores basados en los Grafos de camino clasificados por su precisión en el conjunto de validación.

El mejor resultado para Grafo de caminos se obtuvo con sólo una variación y una diferencia de 2 palabras. El Camino-Los parámetros específicos fueron de *0,3 para cc_th, 0.0001 para centralidad_th y 1 para min_freq. La mejor precisión fue ligeramente superior al 75%,* haciendo de este nuestro mejor clasificador basado en gráficos por un pequeño margen.

En general, los tres tipos de grafos se comportaron de forma similar, obteniendo alrededor de un 75% de precisión en una tarea de detección de sarcasmo muy desafiante y de múltiples clases. Las fig. 13, fig. 14. y fig. 15 muestran las matrices de confusión para los modelos superiores de cada tipo de gráfico.

A partir de las figuras se puede ver cómo todos los métodos funcionaron de manera similar, incluso a nivel de clase. Debido a que los Patrones o caminos y porque creemos que los patrones son más expresivos y se adaptan mejor a los nuevos conjuntos de datos no vistos, elegimos el mejor nuevo conjunto de datos de prueba compuesto principalmente por titulares de noticias, algunos sarcásticos y otros neutrales.

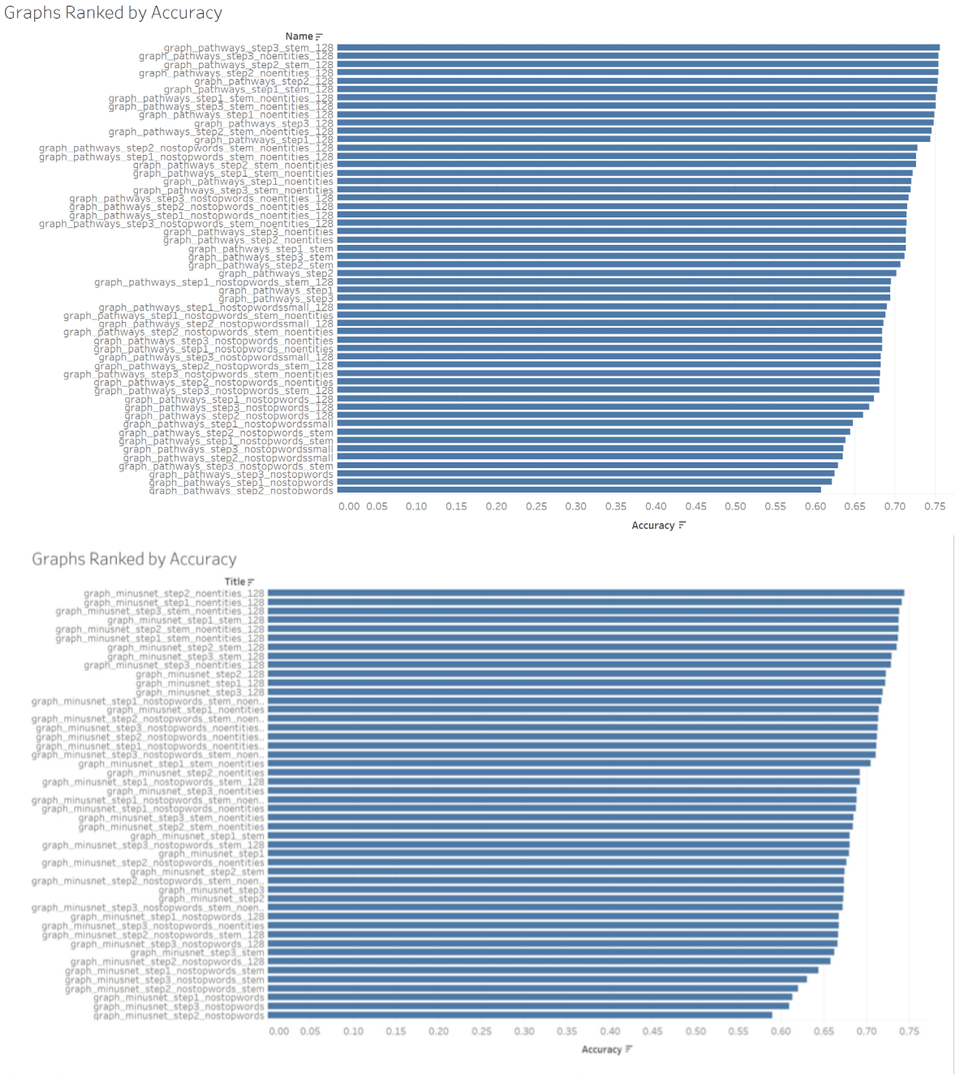


*Fig. 11. Grafos Minusnet- basado en los clasificadores de regresión logística clasificados por su precisión en el conjunto de validación.*



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



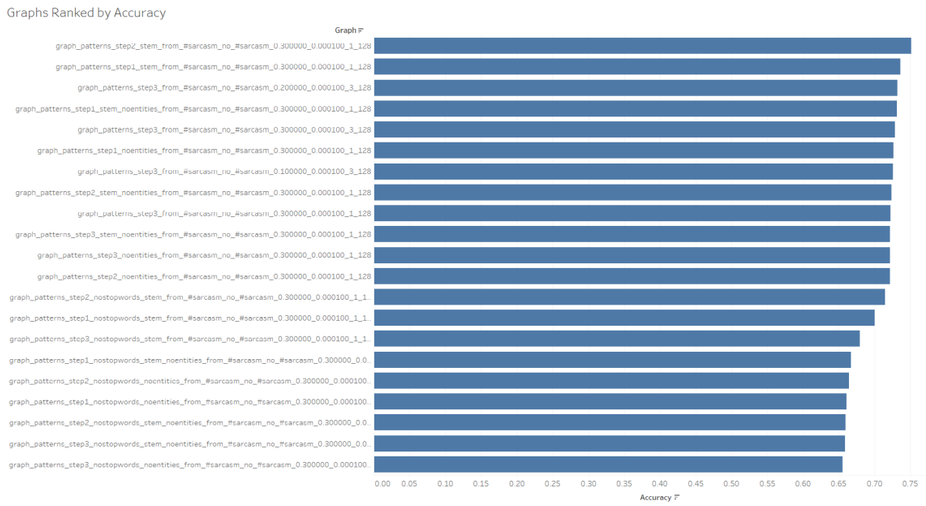


*Fig. 12. Grafos de Patrones- basado en los clasificadores de regresión logística clasificados por su precisión en el conjunto de validación.*

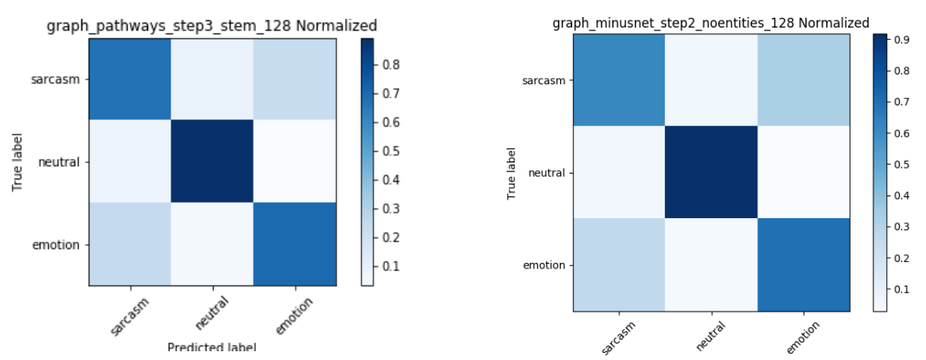


*Fig. 13. Matriz de confusión para el mejor modelo de caminos basado en grafos.*




*Fig. 14. Matriz de confusión para el mejor modelo de caminos basado en grafos de Minusnet.*

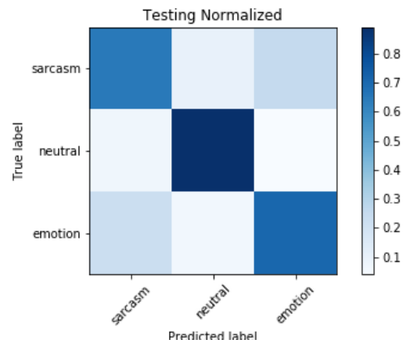


*Fig. 15. Matriz de confusión para el mejor modelo basado en grafos de Patrones.*

*Patterns Graph* clasificador que se comparará en la siguiente sección con otros métodos.

### B. Comparación con otros métodos

En la sección anterior comparamos los tres tipos de grafos propuestos y determinamos que los *Patrones*- se basan en la tecnología son más precisos por un pequeño margen. También encontramos que, en general, el freno ha ayudado a que los tres enfoques funcionen mejor. En la segunda parte, queremos comparar nuestro mejor clasificador con algunas líneas de base y con uno de los más avanzados en detección de emociones y sarcasmo.

El principal aspecto que queremos medir es la capacidad de cada método para adaptarse a datos completamente diferentes de los conjuntos de entrenamiento y validación. Para lograrlo, todos los clasificadores se entrenan y validan con los mismos datos de Twitter, pero se prueban con un nuevo conjunto de datos de prueba compuesto principalmente por titulares de noticias, algunos sarcásticos y otros neutrales.

***1) Conjunto de datos:*** Se utilizaron diferentes conjuntos de datos durante estos experimentos. Los conjuntos de datos se describen a continuación:

- ***Conjuntos de datos de entrenamiento y validación:*** este conjunto de datos está compuesto por el neutral, emociones, y el conjunto de datos de sarcasmo descritos en 4.1.2 y es usado para entrenar el otro método. La idea es que todos los métodos sean entrenados y validadas con la misma data de twitter.

- ***Titulares sarcásticos:*** este dataset consiste de 13,634 titulares de *The Onion. The Onion* es una satírica compañía y periódico de medio digital americano presentando mundanos, eventos cotidianos como noticia de una manera sarcástica. Por ejemplo, Un frustrado CEO admite que Pfizer descubrió la vacuna contra el virus Covid-19 hace meses, pero aún no se pone de acuerdo sobre la campaña publicitaria.

- ***Titulares neutrales:*** este conjunto de datos consiste en 14,985 titulares de *The Huffpost. The Huffpost* es un agregador de noticias y un blog americano.

- ***Tweets de emociones:*** este conjunto de datos contiene 18,018 tweets portadores de emociones, pero son diferentes de los descritos en el experimento anterior.

Como puede verse, el conjunto de datos de las pruebas es significativamente diferente de los conjuntos de entrenamiento y validación. En particular, los textos sarcásticos esta vez son titulares de noticias que son muy diferentes de los tweets sarcásticos. Esto hace que la tarea de clasificación sea mucho más difícil y constituye una muy buena manera de medir el poder de generalización de un clasificador.

***2) Baseline Models (Modelos de línea Base):*** Queremos comparar nuestro mejor enfoque con algunas líneas bases (SOTA) construidas con técnicas avanzadas y de potencia. Las siguientes son las líneas bases utilizadas en este experimento:

- ***TV-C1D-D:*** este modelo está basado en el reciente TensonFlow T*extVectoritation (TV)* que crea capas que luego se alimentan a las redes neuronales para crear clasificadores. *TV-C1D-D* es compuesto de la capa de *TextVectorizer seguido* de la capa de *Convolution1D* y finalmente la *Densa Capa de Red Neuronal.*

- ***TV-BG-TFIDF-D:*** Al igual que el modelo anterior, este también se basa en el *TextVectorizer* pero luego usa una capa de *Bigram TF-IDF* Red Neuronal densamente-conectada como el clasificador.

*Tabla III. Los resultados de la clasificación en el conjunto de pruebas para nuestro mejor enfoque y otros enfoques sota.*

| Model Name | Accuracy |
|---|---|
| **Jammin Patterns Graph** | **46 %** |
| DeepMoji-Finetune-Chain-Thaw | 41 % |
| DeepMoji Finetune-Last | 39 % |
| TV-C1D-D | 38 % |
| DeepMoji-Embs-LogReg | 35 % |
| DeepMoji Finetune-Full | 35 % |
| Sarcasm-RoBERTa | 33 % |
| TV-BG-TFIDF-D | 31 % |

*Tabla IV. Informe de clasificación para el modelo basado en el modelo de grafos de patrones más alto.*

| | *Precision* | *Recall* | *F1 Score* |
|---|---|---|---|
| emociones | 0.78 | 0.45 | 0.57 |
| neutral | 0.50 | 0.76 | 0.60 |
| sarcasmo | 0.16 | 0.15 | 0.16 |
| | | | |
| Promedio Macro | 0.48 | 0.45 | 0.44 |
| Promedio del peso | 0.51 | 0.46 | 0.46 |

***3) Modelos de línea base o estado del arte (SOTA)***

Para validar la afirmación de que nuestro enfoque puede adaptarse a los datos no vistos, mejor que otros enfoques; también queremos comparar nuestro mejor clasificador con algunos de los más recientes y aclamados estudios que han logrado resultados SOTA (*state of the art* - estado del arte).

- ***Sarcasm-RoBERTa:*** Desde el paper U*n enfoque de pre-entrenamiento del BERT robustamente optimizado [36]*, el modelo *RoBERTa* ha ha sido recientemente almacenado en SOTA para una amplia gama de tareas. *Sarcasm-RoBERTa* es una versión afinada y adaptada para la tarea de detección del sarcasmo.

- ***DeepMoji-Embs-LogReg:*** desde el paper *Usando millones de ocurrencias de Emoji para aprender cualquier representación de dominio para detectar el sentimiento, la emoción y el sarcasmo* [31] que nos dio el ampliamente popular modelo *DeepMoji*, implementamos una línea de base y tres de las variaciones que proponen en el documento original para la detección del sarcasmo. *DeepMoji-Embs-LogReg* es la línea de base que trata *DeepMoji* como un extractor de incrustaciones y añade un clasificador de regresión logística superior.

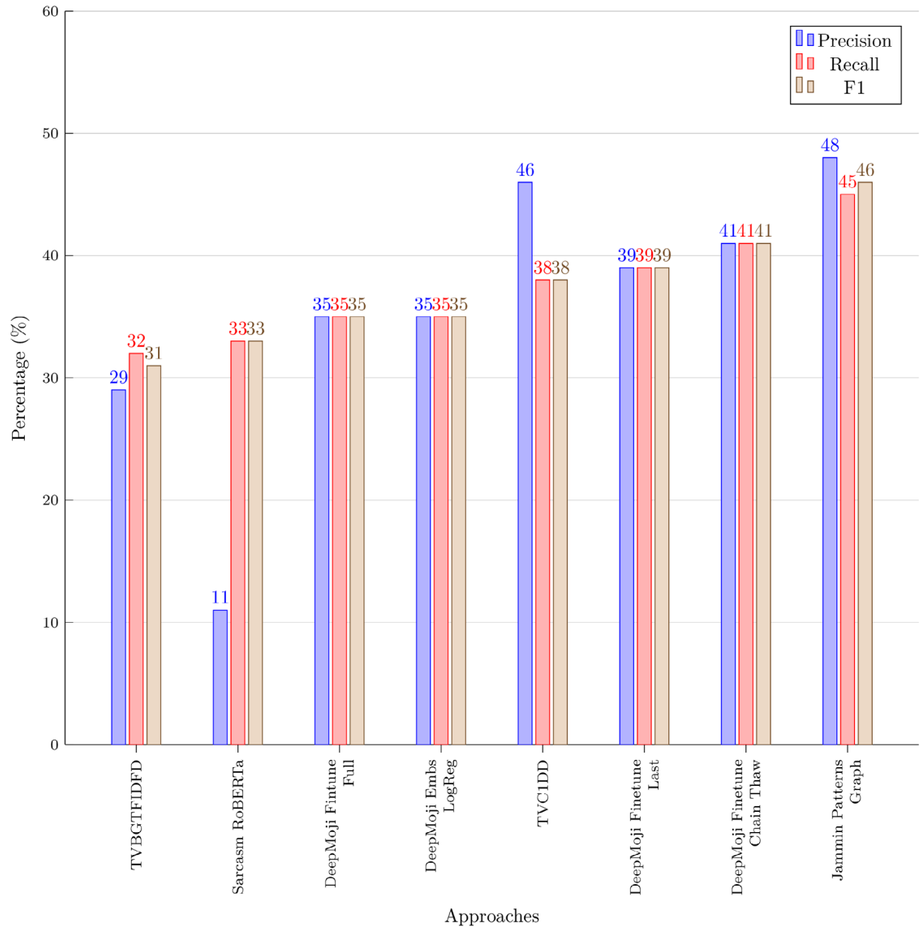


*Fig. 16. Precisión, Recall, y puntajes de F1 para todos los métodos comparados.*

- ***DeepMoji Finetune-Last:*** propuesto en [31], este método perfecciona *DeepMoji* congelando todas las capas excepto la última.

- ***DeepMoji Finetune-Full:*** propuesto en [31], este método afina todas las capas del original *DeepMoji*.

- ***DeepMoji-Finetune-Chain-Thaw:*** propuesto en [31] y también su enfoque de mejor rendimiento, este método afina cada capa por separado.

***4) Configuración del experimento:*** El objetivo de esta segunda ronda de experimentos es demostrar que nuestro enfoque basado en





ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

grafos, para la detección del sarcasmo, puede adaptarse mejor a nuevos conjuntos de datos que otros métodos. Cada una de las líneas de base y métodos propuestos que se compararán con nuestro clasificador de sarcasmo tienen una estructura similar al enfoque propuesto. Todos ellos convierten los textos en incrustaciones y construyen clasificadores encima de eso. Las principales diferencias son el enfoque para extraer las incrustaciones embedding) y el tipo de clasificador. Todos los métodos se entrenan y validan con los conjuntos de datos descritos en 4.1.2. y probado con los conjuntos de datos descritos en 4.2.1.

La tabla III, resume nuestros resultados, basados en nuestros experimentos, con nuestro clasificador *Jammin Patters Graph;* el cual tiene una mayor precisión comparado con los otros clasficadores en la desafiante tarea de detectar el sarcasmo de textos completamente diferentes a los métodos usados para entrenar a los clasificadores. De manera similar, la Fig.16 representa los resultados de nuestros experimentos en cuanto a las medidas de Precisión, Recall y F1 para los demás enfoques, mostrando nuevamente que nuestro enfoque basado en el clasificador de grafos supera a cualquier de los otros métodos. Es particularmente interesante apreciar que nuestro método superó significativamente el mejor metodo en un *5% DeepMoji,* Fine Tuning approach, a saber: *DeepMoji-Finetun e-Chain-Thaw*, que ha sido el método SOTA para la detección de emociones y sarcasmos durante unos años.

Por último, un análisis más profundo del rendimiento de nuestro método se sumariza en la tabla IV en donde se presentan los resultados por clase. En esta tabla se puede distinguir que producto de los resultados de nuestro experimento, la precisión para los textos de emociones presenta valores altos, posiblemente debido al hecho de que el clasificador fue entrenado y probado con tweets del conjunto de dtos de emociones. Del mismo modo, los titulares neutrales tuvieron la puntuación más alta en la F1, posiblemente debido a la similitud entre los titulares de las noticias tradicionales y los titulares de los tweets. Finalmente, y no es sorprendente, que el sarcasmo tuvo el peor desempeño; ya que este es el tipo de datos que fue el más diferente entre el entrenamiento y el conjunto de pruebas, haciendo una tarea; ya aún más difícil de clasificar.

## V. CONCLUSIONES Y TRABAJO FUTURO

En la presente invesitgación, hemos planteado un método novedoso basado en grafos para construir modelos de lenguaje compacto y expresivo para la tarea de detecció n automática del sarcasmo sin tener que obtener un enorme conjunto de datos. En particular, nuestro método logra las siguientes caracterısticas diferenciables a otros métodos:

***1. Independencia del lenguaje:*** si no se utiliza la eliminación de palabras de parada o de contención, las técnicas de generación de grafos y de extracción de incrustaciones (embedding) pueden aplicarse a los textos en cualquier idioma.

***2. Pequeños requisitos de datos:*** utilizando #hashtags para la supervisión a distancia, unos pocos miles de tweets pueden ser agrupados rápidamente para crear los grafos. En comparación con los enfoques como DeepMoji, que necesitó miles de millones de tweets para alcanzar el estatus de SOTA, esto es muy conveniente

***3. No requiere extensivos recursos:*** como no se necesitan grandes conjuntos de datos ni complejas redes de aprendizaje en profundidad, cualquier modelo puede construirse rápidamente con un hardware básico.

***4. Agnóstico de tareas:*** al igual que ser agnóstico del lenguaje, el mismo enfoque también puede aplicarse a otras tareas utilizando un conjunto diferente de hashtags para la supervisión a distancia.

***5.Expresivo:*** el modelo de lenguaje es lo suficientemente expresivo como para ayudar a los clasificadores construidos sobre él. Por lo tanto, puede generalizar mejor que los que tienen enfoques más complejos. En particular, creemos que el uso de patrones puede generalizarse mejor mediante el uso de comodines.

Hemos demostrado experimentalmente que, al encontrar un conjunto de datos experimentales diferentes del conjunto de datos utilizados durante la fase de entrenamiento, nuestro enfoque (Jammin Pattern Graph) supera a otras técnicas avanzadas en un promedio de 5 % en F1, Recall and Precision. Estos resultados se sumarizan en la Fig. 16.

En cuanto al trabajo futuro, nos gustaria probar más combinaciones de parámetros para los graafos basados en patrones.Tambie´n queremos demostrar que nuestro enfoque puede ser utilizado en diferentes idiomas y para diferentes tareas realizando experimentos con otros idiomas como el italiano y el francés. Estos incluyen tareas tan diversas como la clasificación de temas, la detección de discursos de odio, etc.



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021



## REFERENCIAS


[1] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation,” in Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 704–714, 2013.

[2] D. Davidov, O. Tsur, and A. Rappoport, “Semi-supervised recognition of sarcasm in twitter and amazon,” in Proceedings of the fourteenth conference on computational natural language learning, pp. 107–116, 2010.

[3] A. Reyes, P. Rosso, and D. Buscaldi, “From humor recognition to irony detection: The figurative language of social media,” Data & Knowledge Engineering, vol. 74, pp. 1–12, 2012.

[4] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, and A. Reyes, “Semeval-2015 task 11: Sentiment analysis of figurative language in twitter,” in Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp. 470–478, 2015.

[5] C. Liebrecht, F. Kunneman, and A. van Den Bosch, “The perfect solution for detecting sarcasm in tweets# not,” 2013.

[6] F. Barbieri, H. Saggion, and F. Ronzano, “Modelling sarcasm in twitter, a novel approach,” in Proceedings of the 5th Workshop on Computatio- nal Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 50–58, 2014.

[7] G. Abercrombie and D. Hovy, “Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of twitter conversations,” in Proceedings of the ACL 2016 student research workshop, pp. 107–113, 2016.

[8] F. Barbieri, F. Ronzano, and H. Saggion, “Italian irony detection in twit- ter: a first approach,” in The First Italian Conference on Computational Linguistics CLiC-it, vol. 28, 2014.

[9] S. K. Bharti, K. S. Babu, and S. K. Jena, “Parsing-based sarcasm sentiment recognition in twitter data,” in 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1373–1380, IEEE, 2015.

[10] S. Lukin and M. Walker, “Really? well. apparently bootstrapping im- proves the performance of sarcasm and nastiness classifiers for online dialogue,” arXiv preprint arXiv:1708.08572, 2017.

[11] A. Reyes and P. Rosso, “On the difficulty of automatically detecting irony: beyond a simple case of negation,” Knowledge and Information Systems, vol. 40, no. 3, pp. 595–614, 2014.

[12] E. Filatova, “Irony and sarcasm: Corpus generation and analysis using crowdsourcing.,” in Lrec, pp. 392–398, Citeseer, 2012.

[13] K. Buschmeier, P. Cimiano, and R. Klinger, “An impact analysis of features in a classification approach to irony detection in product reviews,” in Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 42–49, 2014.

[14] O. Tsur, D. Davidov, and A. Rappoport, “Icwsm—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews,” in fourth international AAAI conference on weblogs and social media, 2010.

[15] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, and K. Lei, “Sarcasm detection in social media based on imbalanced classification,” in Interna- tional Conference on Web-Age Information Management, pp. 459–471, Springer, 2014.

[16] J. Tepperman, D. Traum, and S. Narayanan, "yeah right": Sarcasm recognition for spoken dialogue systems,” in Ninth international confe- rence on spoken language processing, 2006.

[17] R. Rakov and A. Rosenberg, “sure, i did the right thing”: a system for sarcasm detection in speech.,” in Interspeech, pp. 842–846, 2013.

[18] A. Joshi, V. Tripathi, P. Bhattacharyya, and M. Carman, “Harnessing sequence labeling for sarcasm detection in dialogue from tv series ‘friends’,” in Proceedings of The 20th SIGNLL Conference on Compu- tational Natural Language Learning, pp. 146–155, 2016.

[19] A. Rajadesingan, R. Zafarani, and H. Liu, “Sarcasm detection on twitter: A behavioral modeling approach,” in Proceedings of the eighth ACM international conference on web search and data mining, pp. 97–106, 2015.

[20] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, “Cascade: Contextual sarcasm detection in online discussion forums,” arXiv preprint arXiv:1805.06413, 2018.

[21] D. Bamman and N. A. Smith, “Contextualized sarcasm detection on twitter,” in Ninth International AAAI Conference on Web and Social Media, 2015.

[22] A. Joshi, V. Sharma, and P. Bhattacharyya, “Harnessing context in-congruity for sarcasm detection,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 757–762, 2015.
[23] B. C. Wallace, E. Charniak, et al., “Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1035– 1044, 2015.
[24] Z. Wang, Z. Wu, R. Wang, and Y. Ren, “Twitter sarcasm detection exploiting a context-based model,” in international conference on web information systems engineering, pp. 77–91, Springer, 2015.
[25] A. Joshi, P. Jain, P. Bhattacharyya, and M. Carman, “Who would have thought of that!’: A hierarchical topic model for extraction of sarcasm-prevalent topics and sarcasm detection,” arXiv preprint ar- Xiv:1611.04326, 2016.
[26] R. Gonza´lez-Ibánez, S. Muresan, and N. Wacholder, “Identifying sarcasm in twitter: a closer look,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 581–586, 2011.
[27] A. Reyes and P. Rosso, “Making objective decisions from subjective data: Detecting irony in customer reviews,” Decision support systems, vol. 53, no. 4, pp. 754–760, 2012.
[28] R. Kreuz and G. Caucci, “Lexical influences on the perception of sarcasm,” in Proceedings of the Workshop on computational approaches to Figurative Language, pp. 1–4, 2007.
[29] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis, “A unified deep learning architecture for abuse detection,” in Proceedings of the 10th ACM Conference on Web Science, pp. 105– 114, 2019.
[30] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,” arXiv preprint arXiv:1708.00524, 2017.
[31] A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, and M. Carman, “Are word embedding-based features useful for sarcasm detection?,” arXiv preprint arXiv:1610.00883, 2016.
[32] D. Paranyushkin, “Identifying the pathways for meaning circulation using text network analysis,” Nodus Labs, vol. 26, 2011.
[33] C. Argueta, E. Saravia, and Y.-S. Chen, “Unsupervised graph-based pat- terns extraction for emotion classification,” in Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, pp. 336–341, 2015.
[34] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701–710, 2014.
[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.
[36] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021

# AUTHORS

## Axel Rodríguez-García

Axel Rodríguez nació en Panamá. Recibió el título de Licenciado en Ingeniería de Sistemas Computacionales en la Universidad Tecnológica de Panamá, Panamá, en 1999; y una Maestría en Ciencias del Departamento de Ingeniería Industrial en la Universidad de Lousiville, en Los Estados Unidos, en el año 2000. Actualmente es candidato a doctor en el programa de doctorado en Ingeniería de Proyectos en la Universidad Tecnológica de Panamá, Panamá.

## Armando Jipsion

Armando Jipsion nació en Panamá. Recibió el título de Licenciado en Tecnología en Programación y Análisis de Sistemas en la Universidad Tecnológica de Panamá, 1988. Obtuvo el Técnico en Ingeniería con Especialización en Programación y Análisis de Sistemas Universidad Tecnológica de Panamá, 1988. Obtuvo el Doctorado en Ingeniería de Proyectos de la Universidad Tecnológica de Panamá, 2012. La Maestría en Administración de Sistemas de Información en la Universidad Santa María La Antigua, 1995.
Tiene 30 años de ser Profesor Regular Titular.

A. Rodríguez-García and A. Jipsion, "Modelos de grafos para la detección de datos de texto no estructurados como el sarcasmo", Latin-American Journal of Computing (LAJC), vol. 8, no. 1, 2021.



ISSN:1390-9266 e-ISSN:1390-9134 LAJC 2021