

Análisis de Sentimientos de la serie de televisión Game Of Thrones utilizando Node.js y Twitter

27 de junio de 2019

UNIVERSIDAD NACIONAL DE LA RIOJA

SEDE CAPITAL

LIC. EN SISTEMAS DE INFORMACIÓN

ANL. SIST. MARCOS J. PEÑA POLLASTRI

Índice

1. Resumen	4
2. Temas	5
3. Problema	5
4. Marco Teórico	5
4.1. Procesamiento de Lenguaje Natural	5
4.1.1. Aplicaciones del Procesamiento de Lenguaje Natural	6
4.2. Análisis de sentimientos	6
4.2.1. Tipos de análisis de sentimiento	7
4.2.2. Análisis a nivel de documento	7
4.3. La red social Twitter	8
4.4. Node.JS	8
4.4.1. Descripción general	9
4.4.2. Arquitectura de la plataforma	9
4.4.3. Historia	10
4.5. Express	11
4.5.1. ¿Qué popularidad tiene Node/Express?	11
4.5.2. ¿Es Express dogmático?	12
4.6. API de Twitter	12
4.6.1. Objeto Tweet	13
4.7. Sentiment	13
4.7.1. Funcionamiento	13
5. Identificación de variables	13
6. Objetivo general	14
7. Objetivos específicos	14
8. Hipótesis	14
9. Diseño metodológico de la investigación	14
9.1. Según la finalidad	14
9.2. Según la dimensión temporal	14
9.3. Según la lógica	14
9.4. Según la direccionalidad	14
9.5. Unidad de análisis	14
9.6. Unidad de observación	15
9.7. Universo	15
9.8. Muestra	15
9.9. Criterio de selección	15

I. Análisis de los datos	15
II. Conclusión	16
10.Conclusión	16
III. Bibliografía	16

1. Resumen

.

.

.

2. Temas

1. Procesamiento Natural del Lenguaje.
2. Análisis de sentimientos.
3. Sistemas web basados en eventos.
4. Bases de datos no relacionales.

3. Problema

¿Qué sentimientos se generan durante la emisión de un capítulo de la serie de televisión Game Of Thrones en los usuarios de la red social Twitter de habla inglesa, durante el año 2019?

4. Marco Teórico

4.1. Procesamiento de Lenguaje Natural

El procesamiento del lenguaje natural PLN, o NLP del idioma inglés Natural Language Processing, es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio del lenguaje natural, es decir, de las lenguas del mundo. Por medio de estos algoritmos o procesos matemáticos traducen datos (lenguaje natural) en el cerebro de la máquina para que éste determine patrones y pueda generar una respuesta. Mucho dependerá del tipo, calidad y cantidad de datos de entrenamiento para determinar el éxito de respuesta de la máquina. Por ejemplo, para entrenar a Google Home, el tipo de datos que se utilizó fue la voz humana y seguramente con una extensa variedad de tonalidades y registros de voz (calidad y cantidad) provenientes de hombres, mujeres, niños, adultos, extranjeros y locales.

El PLN no trata de la comunicación por medio de lenguas naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente, que se puedan realizar por medio de programas que ejecuten o simulen la comunicación. Los modelos aplicados se enfocan no solo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos humanos y a la organización de la memoria. El lenguaje natural sirve solo de medio para estudiar estos fenómenos. Hasta la década de 1980, la mayoría de los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en PLN con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.

4.1.1. Aplicaciones del Procesamiento de Lenguaje Natural

Las aplicaciones del PLN son muy variadas, ya que su alcance es muy grande, algunas de las aplicaciones son:

- Traducción automática.
- Recuperación de la información.
- Extracción de Información y resúmenes.
- Resolución cooperativa de problemas.
- Tutores inteligentes.
- Reconocimiento de Voz.

4.2. Análisis de sentimientos

El análisis de sentimientos es una tarea incluida en el ámbito del Procesamiento de Lenguaje Natural o NLP (Natural Language Processing). Su objetivo es encontrar contenido subjetivo en los textos de entrada. El análisis de sentimientos busca extraer opiniones y la polaridad de éstas, mediante la identificación de características que determinan cuán positivo o negativo es el texto.

El análisis de sentimientos ha sido considerado inicialmente como una subdisciplina de clasificación basada en la opinión. La técnica en general puede estar orientada a tres grandes tipos de tareas:

- Clasificación de sentimientos: realizar una clasificación de un conjunto de opiniones en tres categorías: positivas, negativas o neutrales. Presenta desafíos adicionales el hecho de que las opiniones se encuentran en múltiples idiomas o provienen de varios dominios, como biología, sociología, etc.
- Clasificación de subjetividad: determinar si una oración es subjetiva u objetiva. Una oración objetiva contiene información imparcial, mientras que una oración subjetiva contiene información de carácter personal como opiniones.
- Resumen de opinión: permitir extraer las características principales que son compartidas por uno o más documentos y el sentimiento acerca de estas características.

En el análisis de redes sociales, particularmente en análisis de sentimientos a partir del estudio de tweets para obtener su polaridad, existen dos métodos que son los más populares. El primero de ellos utiliza el enfoque del aprendizaje computacional (machine-learning approaches) y el segundo utiliza diccionarios léxicos.

El aprendizaje computacional analiza la información automáticamente de forma supervisada, basándose en conjuntos de entrenamiento que son utilizados para catalogar al resto de las opiniones encontradas en la web, realizando pruebas y luego validandose. Las principales técnicas de este método son: Support Vector Machines (SVM), Naive Bayes y

Clasificadores de Máxima Entropía. En estas técnicas se utiliza la categoría gramatical de las palabras, la presencia y frecuencia de algunos términos y su composición semántica. Sin embargo, la mayoría de estos métodos son acompañados de algún diccionario que entrega información a priori de los términos para obtener las polaridades respectivas. En algunos casos, estos diccionarios son realizados por personas y en otros se ocupa un sistema semiautomático. El método de diccionarios léxicos se basa en una lista de palabras con un determinado peso y/o categoría emocional. Estos diccionarios presentan principalmente adjetivos, que son los que aportan mayor información al momento de analizar los sentimientos, aunque también incluye verbos, adverbios y sustantivos.

La mayor cantidad de versiones de diccionarios está en idioma inglés aunque existen algunas en español, pero en general en versión beta. Éstos permiten determinar si una frase es negativa o positiva dependiendo de la cantidad de palabras presentes en el diccionario y de la fuerza de su sentimiento. Uno de los diccionarios más completos es LIWC, presente una versión en inglés bastante completa y una en español que está aún en etapa de desarrollo. En este caso, las palabras son etiquetadas en una categoría determinada y además se les asigna una ponderación.

4.2.1. Tipos de análisis de sentimiento

A la hora de extraer esta información, hay una gran variedad de métodos y algoritmos dependiendo del nivel de granularidad del análisis que queramos llevar a cabo. Se distinguen tres niveles: nivel de documento, de oración o de aspecto.

El análisis a nivel de documento determina el sentimiento general expresado en un texto, mientras que el análisis a nivel de frase lo especifica para cada una de las oraciones del texto. Sin embargo, estos dos tipos de análisis no profundizan en detalle el elemento que a las personas les gusta o no. No especifican sobre qué es la opinión, ya que considerando la opinión general de un objeto como positiva (o negativa) no significa que el autor tenga una opinión positiva (o negativa) de todos los aspectos de dicho objeto.

Para este trabajo nos enfocamos en realizar un análisis a nivel de documento, ya que, debido al límite en los mensajes, los autores suelen ser concisos y van directo al grano sin tener la posibilidad de incluir varios aspectos diferenciados en un solo tweet. Por esta razón, usar el tweet como unidad de análisis parece proveer un nivel de granularidad adecuado para hacer un análisis de sentimiento desglosado.

4.2.2. Análisis a nivel de documento

Considerada como una de las tareas más simples de la minería de opiniones, el análisis a nivel de documento apunta a clasificar la opinión de un documento, en este caso un tweet. Esta tarea no considera los detalles en cuanto a entidades o aspectos, sino que considera el documento como un todo, el cual será etiquetado como positivo o negativo. Esta puede ser considerada como una tarea tradicional de clasificación de texto, donde las clases son las diferentes orientaciones en cuanto a los sentimientos. No obstante, para asegurar que este tipo de análisis tenga sentido asumimos que cada documento expresa una única opinión sobre una única entidad. Si bien esto puede parecer una limitación,

porque en un tweet uno podría expresar más de una opinión hacia distintas entidades, en la práctica produce resultados positivos, ya que los usuarios suelen enfocarse en un único aspecto en cada tweet. Seguramente en otros contextos, o si no estuviera limitado el largo de los mensajes, sería una buena idea considerar sistemas de análisis más complejos que permitan realizar un análisis con mayor granularidad. Las palabras que conforman las opiniones son el factor determinante en el análisis de sentimientos también es una buena opción utilizar métodos de aprendizaje basados en el uso de lexicones. Estos son diccionarios que contienen listados de palabras etiquetadas con el sentimiento asociado correspondiente, en algunos casos por un valor que también indica la intensidad del mismo. Luego, dada una opinión, simplemente podemos buscar el valor de todas las palabras que la componen y sumar el valor asociado para finalmente clasificar la opinión como positiva o negativa según corresponda.

4.3. La red social Twitter

Twitter se creó en 2006 como una red social en base a contenido con forma de SMS. Desde entonces ha crecido rápidamente ganando mucha popularidad en los últimos años. Según los datos oficiales que figuran en su web son aproximadamente 326 millones de usuarios activos por mes que acceden diariamente para compartir experiencias y opiniones convirtiéndose así en una herramienta ideal para la realización de encuestas y sondeos.

Twitter permite a los usuarios enviar y leer mensajes de texto de hasta 140 caracteres, conocidos como tweets. Este sitio es una gran fuente de información subjetiva en tiempo real ya que estos millones de usuarios comparten opiniones sobre diferentes aspectos de su vida cotidiana. Twitter ha sido usada para una variedad de propósitos en diferentes industrias y situaciones. Por ejemplo, los usuarios pueden encontrar o emitir opiniones sobre un producto o servicio de su interés, las compañías y figuras públicas pueden controlar su reputación en línea, se puede conocer la opinión de cada usuario respecto a implementación de políticas públicas, campañas de difusión, entretenimiento, entre otras.

Twitter permite identificar estos temas a través de los denominados hashtag o etiquetas, que se caracterizan por comenzar con el carácter # y una cadena de caracteres a continuación formada por una o varias palabras concatenadas. Esta etiqueta funciona como metadato y permite que los tweets se añadan a una lista de mensajes con hashtag similares. Esto permite a los usuarios obtener resultados rápidamente sobre un mismo tema.

4.4. Node.JS

Node.Js o también conocido como Node, es un entorno de ejecución de Javascript en backend multiplataforma, código abierto. Esto permite la ejecución de código Javascript fuera del navegador web, y poder escribir programas de líneas de comando o scripting para servidores (como por ejemplo la creación de paginas web dinámicas).

Sumado a esto, Node.js representa el paradigma “Javascript en todos lados”, unificando el desarrollo de las aplicaciones web en un solo lenguaje, en vez de utilizar diferentes lenguajes para el scripting en servidor o en el cliente.

Node.js posee una arquitectura orientada a los eventos, capaz de realizar entradas y salidas asíncronas. Esto permite optimizar el consumo y escalabilidad de aplicaciones web en muchas operaciones de E/S, como también aplicaciones en tiempo real.

El proyecto de desarrollo distribuido de Node.js, a manos de la Node.js Foundation, es facilitada por el programa de proyectos colaborativos de la Linux Foundation. Algunas de las grandes empresas que utilizan software construido en Node.js son: GoDaddy, Groupon, IBM, LinkedIn, Microsoft, Netflix, PayPal, Walmart, entre otros.

4.4.1. Descripción general

Node.js permite la creación de servidores web y herramientas de red utilizando JavaScript y una colección de "módulos" que manejan varias funciones centrales. Se proporcionan módulos para la E / S del sistema de archivos, redes (DNS, HTTP, TCP, TLS / SSL o UDP), datos binarios (buffers), funciones de criptografía, flujos de datos, y otras funciones básicas. Los módulos de Node.js utilizan una API diseñada para reducir la complejidad de escribir aplicaciones de servidor.

Aunque inicialmente el sistema de módulos se basaba en el patrón de módulos `commonjs`, la reciente introducción de módulos en la especificación ECMAScript ha cambiado la dirección de usar módulos ECMAScript en Node.js de forma predeterminada.

Node.js es oficialmente compatible con Linux, macOS y Microsoft Windows 7 y Server 2008 (y posteriores) por parte del equipo. También hay soporte de nivel 2 para SmartOS e IBM AIX. Y soporte experimental para FreeBSD; OpenBSD también funciona y, por ejemplo, las versiones LTS disponibles para IBM i (AS/400). El código fuente provisto también puede construirse en sistemas operativos similares a los que son oficialmente compatibles o ser modificado por terceros para que sean compatibles con otros, como los servidores NonStop y Unix. Alternativamente, se puede escribir con CoffeeScript (una alternativa de JavaScript), Dart o TypeScript (formas de JavaScript fuertemente tipadas), o cualquier otro lenguaje que se pueda compilar en JavaScript.

Node.js se utiliza principalmente para crear programas de red como los servidores web. La diferencia más significativa entre Node.js y PHP es que la mayoría de las funciones en el bloque PHP hasta su finalización (los comandos solo se ejecutan después de que terminan los comandos anteriores), mientras que las funciones de Node.js no se bloquean (los comandos se ejecutan simultáneamente o incluso en paralelo, y use devoluciones de llamada para indicar la finalización o el fallo).

4.4.2. Arquitectura de la plataforma

Node.js lleva la programación dirigida por eventos a los servidores web, lo que permite el desarrollo de servidores web rápidos en JavaScript. Los desarrolladores pueden crear servidores escalables sin utilizar subprocesos, mediante el uso de un modelo simplificado de programación dirigida por eventos que utiliza devoluciones de llamada para indicar la finalización de una tarea. Node.js conecta la facilidad de un lenguaje de scripting (JavaScript) con el poder de la programación en red de Unix.

Node.js fue construido en el motor de JavaScript Google V8 ya que fue de código

abierto bajo la licencia BSD . Es competente con los fundamentos de Internet como HTTP, DNS, TCP. JavaScript también era un lenguaje muy conocido, lo que hacía que Node.js fuera accesible para la comunidad de desarrollo web.

4.4.3. Historia

Node.js fue escrito inicialmente por Ryan Dahl en 2009, aproximadamente trece años después de la introducción del primer entorno de JavaScript del lado del servidor, Live-Wire Pro Web de Netscape. La versión inicial solo era compatible con Linux y Mac OS X. Su desarrollo y mantenimiento fue dirigido por Dahl y, posteriormente, patrocinado por Joyent .

Dahl criticó las posibilidades limitadas del servidor web más popular en 2009, el Servidor HTTP Apache , para manejar muchas conexiones concurrentes (hasta 10,000 o más) y la forma más común de crear código (programación secuencial).

Dahl demostró el proyecto en la JSConf europea inaugural el 8 de noviembre de 2009. Para ello combinó el motor de JavaScript V8 de Google , un ciclo de eventos y una API de E/S de bajo nivel.

En enero de 2010, se introdujo un administrador de paquetes para el entorno Node.js llamado npm. El administrador de paquetes facilita a los programadores publicar y compartir el código fuente de las bibliotecas Node.js y está diseñado para simplificar la instalación, actualización y desinstalación de las bibliotecas.

En junio de 2011, Microsoft y Joyent implementaron una versión nativa para Windows de Node.js. La primera compilación de Node.js compatible con Windows se lanzó en julio de 2011.

En enero de 2012, Dahl se hizo a un lado, promoviendo al compañero de trabajo y creador de npm Isaac Schlueter para gestionar el proyecto. En enero de 2014, Schlueter anunció que Timothy J. Fontaine lideraría el proyecto.

En diciembre de 2014, Fedor Indutny comenzó io.js, una bifurcación de Node.js. Debido al conflicto interno sobre el liderazgo de Joyent, io.js se creó como una alternativa de gobierno abierto con un comité técnico independiente. A diferencia de Node.js, los autores planeaban mantener io.js actualizado con las últimas versiones del motor de JavaScript Google V8.

En febrero de 2015, se anunció la intención de formar una Fundación Node.js neutral. Para junio de 2015, las comunidades Node.js y io.js votaron para trabajar juntas bajo la Fundación Node.js.

En septiembre de 2015, Node.js v0.12 y io.js v3.3 se fusionaron de nuevo en Node v4.0. Esta combinación trajo las características del V8 ES6 a Node.js y un ciclo de lanzamiento de soporte a largo plazo. A partir de 2016, el sitio web io.js recomienda que los desarrolladores vuelvan a Node.js y que no se planifiquen más lanzamientos de io.js debido a la fusión.

4.5. Express

Express.js, o simplemente Express, es un framework de desarrollo de aplicaciones web para Node.js. Fue liberado como software de código abierto con la Licencia MIT. Está diseñado para la construcción de aplicaciones web y APIs. Ha sido llamado como el framework de servidor estándar de facto para Node.js.

El autor original, TJ Holowaychuk, lo describió como un servidor inspirado en Sinatra. Esto significa que es minimalista, y algunas de las varias funciones se encuentran disponibles a través de plugins.

4.5.1. ¿Qué popularidad tiene Node/Express?

La popularidad de un framework web es importante porque es un indicador de si se continuará manteniendo y qué recursos tienen más probabilidad de estar disponibles en términos de documentación, librerías de extensiones y soporte técnico.

No existe una medida disponible de inmediato y definitiva de la popularidad de los frameworks de lado servidor (aunque sitios como Hot Frameworks intentan asesorar sobre popularidad usando mecanismos como contar para cada plataforma el número de preguntas sobre proyectos en GitHub y StackOverflow). Una pregunta mejor es si Node y Express son lo "suficientemente populares" para evitar los problemas de las plataformas menos populares. ¿Continúan evolucionando? ¿Puedes conseguir la ayuda que necesitas? ¿Hay alguna posibilidad de que consigas un trabajo remunerado si aprendes Express?

De acuerdo con el número de compañías de perfil alto que usan Express, el número de gente que contribuye al código base, y el número de gente que proporciona soporte tanto libre como pagado, podemos entonces decir que Express es un framework popular.

Express proporciona mecanismos para:

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.

A pesar de que Express es en sí mismo bastante minimalista, los desarrolladores han creado paquetes de middleware compatibles para abordar casi cualquier problema de desarrollo web. Hay librerías para trabajar con cookies, sesiones, inicios de sesión de usuario, parámetros URL, datos POST, cabeceras de seguridad y muchos más. Puedes encontrar una lista de paquetes middleware mantenida por el equipo de Express en Express Middleware (junto con una lista de algunos de los paquetes más populares de terceros).

4.5.2. ¿Es Express dogmático?

Los frameworks web frecuentemente se refieren a sí mismos como "dogmáticos" ("opinionated") o "no dogmáticos" ("unopinionated").

Los frameworks dogmáticos son aquellos que opinan acerca de la "manera correcta" de gestionar cualquier tarea en particular. Ofrecen soporte para el desarrollo rápido en un dominio en particular (resolver problemas de un tipo en particular) porque la manera correcta de hacer cualquier cosa está generalmente bien comprendida y bien documentada. Sin embargo pueden ser menos flexibles para resolver problemas fuera de su dominio principal, y tienden a ofrecer menos opciones para elegir qué componentes y enfoques pueden usarse.

Los frameworks no dogmáticos, en contraposición, tienen muchas menos restricciones sobre el modo mejor de unir componentes para alcanzar un objetivo, o incluso qué componentes deberían usarse. Hacen más fácil para los desarrolladores usar las herramientas más adecuadas para completar una tarea en particular, si bien al coste de que necesitas encontrar esos componentes por uno mismo.

Express es no dogmático, transigente. Puede insertar casi cualquier middleware compatible que se desee dentro de la cadena de manejo de la petición, en casi cualquier orden que se quiera. Permite estructurar una app en un fichero o múltiples ficheros y usar cualquier estructura de directorios.

4.6. API de Twitter

Una interfaz de programación de aplicaciones (API) es un conjunto de definiciones de subrutinas, protocolos de comunicación y herramientas para crear software. En términos generales, es un conjunto de métodos de comunicación claramente definidos entre varios componentes. Una buena API hace que sea más fácil desarrollar un software al proporcionar todos los bloques de construcción, que luego son ensamblados por el programador .

Una API puede ser para un sistema basado en web, sistema operativo , sistema de base de datos, hardware de computadora o biblioteca de software .

En el caso de la API de Twitter, estamos ante la presencia de una API Web, en el cual las consultas son hechas a través de solicitudes GET o POST a la misma.

Entre las funciones que proporciona la API de Twitter para los desarrolladores tenemos:

1. Integrar Twitter a nuestra aplicación web.
2. Twittear o retwittear desde nuestra aplicación automáticamente.
3. Realizar campañas de publicidades.
4. Capturar tweets en tiempo real, o streaming.
5. Búsqueda de tweets históricos (limitado para usuarios gratuitos).

4.6.1. Objeto Tweet

Los tweets son el bloque de construcción atómico básico de todas las cosas de Twitter. Los tweets son también conocidos como “actualizaciones de estado.” El objeto Tweet tiene una larga lista de ‘nivel raíz’ atributos, incluidos los atributos fundamentales tales como id, created_at, y text. Los objetos de tweet son también el objeto "padre" de varios objetos hijos. Los objetos anidados incluyen user, entities y extended_entities. Los tweets que están etiquetados geográficamente tendrán un objeto place secundario.

4.7. Sentiment

Sentiment es un paquete de software alojado en NPM que permite el análisis de sentimientos de cadena de textos utilizando el listado de palabras AFINN-165 y el Emoji World Sentiment Ranking.

Entre las características principales que lo destacan es:

1. Licencia MIT.
2. Análisis de sentimiento en ingles.
3. Posibilidad de agregar nuevos lenguajes.
4. Opciones de benchmark.

4.7.1. Funcionamiento

A la cadena de texto a analizar, Sentiment aplica un proceso de tokenización. Esto significa dividir el conjunto de caracteres en “tokens”, es decir, subcadenas de texto formadas a partir de separar la cadena en los espacios, y eliminando los caracteres no alfanuméricos.

Cada uno de los tokens obtenidos es comparado con su valor en el diccionario AFFIN-165 y el Emoji World Sentiment Ranking. En base a ello, cada token obtiene un valor de sentimiento, que es un valor numérico entre -5 y 5. La suma de los valores de sentimientos corresponden al puntaje S:

$$S = \sum_1^n t_1 + t_2 + \dots + t_n$$

Y siendo el valor comparativo C respecto a la cantidad de tokens:

$$C = \sum_1^n \frac{t_1+t_2+\dots+t_n}{n}$$

5. Identificación de variables

- Variable de estudio: sentimientos generados por la serie de televisión Game Of Thrones.
- Variable disciplinar: Cálculo de los sentimientos generados por la serie Game Of Thrones a través de un proceso algorítmico ejecutado en un servidor web.

6. Objetivo general

Capturar, calcular, almacenar y describir los sentimientos generados durante la emisión de un episodio de Game Of Thrones por usuarios de Twitter de habla inglesa.

7. Objetivos específicos

1. Identificar los mensajes enviados a modo de tweet que sean relacionados con Game Of Thrones.
2. Capturar esos mensajes, y posteriormente calcular su polaridad de sentimiento.
3. Almacenar los datos generados en una base de datos.
4. Inferir sobre los sentimientos generados en base a los resultados obtenidos.

8. Hipótesis

Los sentimientos generados por la serie Game Of Thrones serán en su mayoría de polaridad positiva debido a la gran popularidad de la misma.

9. Diseño metodológico de la investigación

9.1. Según la finalidad

Investigación de finalidad descriptiva, debido a que se describen y clasifican los sentimientos generados por la emisión de un capítulo de una serie de televisión.

9.2. Según la dimensión temporal

Investigación transversales, se realiza una única medición en un momento determinado por medio de un software desarrollado específicamente para dicha finalidad.

9.3. Según la lógica

Investigación cuantitativa, se explica el fenómeno haciendo una interpretación de datos numéricos.

9.4. Según la direccionalidad

Investigación ????, los hechos ocurren en el momento de la investigación.

9.5. Unidad de análisis

Mensajes enviados a la red social Twitter que sean referidos a la serie de televisión Game Of Thrones.

9.6. Unidad de observación

Usuarios de la red social Twitter que envíen mensajes durante la emisión del capítulo.

9.7. Universo

Todas las personas que se encuentren viendo el capítulo de la serie al momento de la emisión.

9.8. Muestra

58.020 (cincuenta y ocho mil veinte) tweets.

9.9. Criterio de selección

Aquellos tweets que incluyen la palabra "#gameofthrones" capturados entre las 22:00 y 23:00 hs del día 21 de abril del 2019.

Parte I.

Análisis de los datos

Durante la hora de duración del episodio se han capturado 58020 tweets con el hashtag #Gameofthrones.

De los cuales:

- 27972 tienen un valor de sentimientos neutro (igual a 0).
- 30048 tienen un valor de sentimientos no neutro (distinto de 0).
 - 15710 tienen un valor de sentimientos positivo.
 - 14338 tienen un valor de sentimientos negativo.

En un diagrama de torta, esto puede ser graficado por porcentajes de la siguiente manera:

////////DIAGRAMA////////

Tomando los intervalos para clasificar de mejor manera los sentimientos siendo:

1. Muy negativos entre -5 y -3.
2. Negativos entre -3 y 0.
3. Neutro 0.
4. Positivo entre 0 y 3.
5. Muy positivos entre 3 y 5.

////////DIAGRAMA BARRAS////////

Podemos apreciar que la distribución de sentimientos corresponde a lo que uno esperaríamos, siendo los extremos poco frecuentes, y elevando la misma en los valores neutros.

Parte II.

Conclusión

10. Conclusión

.....

Parte III.

Bibliografía

1. IBM Big Data & Analytics Hub. <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>
2. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2 (2008), no. 1-2, 1–135.
3. Fawcett, Robin P. y Gordon H. tucker, 1990: “Demonstration of GenesYs: a very large, semantically based systemic functional grammar” en Proceedings of the 13th International Conference on Computational Linguistics, Helsinki, 47-49.
4. Ferrari, Giacomo, 2004: “state of the art in computational linguistics” en Piet van sterkenBurg (ed.): Linguistics today: facing a greater challenge, ámsterdam-Filadelfia: John Benjamins, 163-186.
5. Fillmore, charles J., 1968: “The case for case” en emmon W. Bach y Robert T. harms (ed.): Universals in linguistic theory, nueva York: Holt, Rinehart and Winston, 1-88.
6. <https://developer.twitter.com/en/docs.html>
7. <https://docs.npmjs.com/>
8. <https://www.npmjs.com/package/sentiment>
9. <https://www.npmjs.com/package/twitter>
10. <https://docs.mongodb.com/>
11. <https://mongoosejs.com/docs/> <https://expressjs.com/en/4x/api.html>