

Análisis y clasificación de tweets contextuales de desastres

Tania Alcántara, Omar García-Vázquez,
Hiram Calvo, Marco A. Cardoso-Moreno

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Ciencias Cognitivas Computacionales,
México

{talcantaram2020, hcalvo,
mcardosom2021}@cic.ipn.mx, omar.gava@hotmail.com

Resumen. El despliegue de información de los desastres ha ido cambiando, hoy en día esta divulgación de información se hace de manera más rápida a través de las redes sociales, especialmente por Twitter, pero, ya que Twitter es completamente abierto, hace que el posteo de información lo haga cualquier usuario y la confianza se pierda. Este trabajo se centra en el análisis y clasificación de un conjunto de textos etiquetados como desastre y no desastre, donde los etiquetados como no desastre incluyen contexto metafórico. La clasificación va centrada en modelos clásicos, como Random Forest, Support Vector Machine, Decision Tree y XGBOOST, esto con los extractores de características SentenceBert y n-gramas y así, determinar la importancia de la selección de características para encontrar relaciones entre los textos y la importancia de las palabras.

Palabras clave: Aprendizaje automático, clasificación, procesamiento de lenguaje natural, N-gramas, SentenceBERT, desastres

Analysis and Classification of Contextual Disaster Tweets

Abstract. The shipping of information on disasters has changed over time, nowadays, social networks allow for faster information disclosure about these topics, specially Twitter; since this platform is completely open, i.e., there is little to none restriction on what a user can post, hence, creating a lack of confidence and trust on the information available. This paper focuses on the analysis and classification of a set of texts labeled as disaster and non-disaster, where those labeled as non-disaster include metaphorical context. The classification is focused on classical models, such as Random Forest, SVM, Decision Trees and XGBOOST with different extractors, with the help of Sentence Bert feature extraction and n-grams, and thus determine the importance of feature selection to find relationships between texts and the importance of words.

Keywords: Machine learning, classification, natural language processing, N-grams, SentenceBERT, disasters.

1. Introducción

La huella del cambio climático ha impactado en la forma en la que nuestro planeta se comporta y esto ha provocado un aumento de las catástrofes. De acuerdo a la Organización Meteorológica Mundial (OMM), los desastres naturales han aumentado en los últimos 50 años, además el Atlas de Mortalidad y Pérdidas Económicas por Fenómenos Meteorológicos, Climáticos e Hídricos de los organismos, reportan que entre 1970 y 2019, los desastres naturales representan el 50 % de todos los desastres, el 74 % de las pérdidas económicas y el 45 % de las muertes [1].

Así mismo, y derivado de la forma en la que vivimos, no solo encontramos desastres naturales, sino humanos, como choques automovilísticos, incendios provocados, entre algunos otros.

Hace 30 años, cuando un evento de esa magnitud ocurría, la información era desplegada principalmente a través de los medios tradicionales, como la televisión, la radio, y en algunos casos, debido a los plazos para la impresión de papel, era necesario esperar hasta el siguiente día para conocer la información a través de periódicos y revistas.

Además, los medios oficiales del gobierno lanzaban comunicados donde se indicaban las acciones a seguir, la ayuda y los datos de más relevancia. La única manera en que la gente cercana al problema pudiera dar información, era a través de la radio por medio de los radioaficionados, los cuales proporcionaban servicios de comunicación críticos en momentos de crisis [2].

Hoy en día, derivado de los alcances tecnológicos, los radioaficionados han evolucionado en el entorno de las redes sociales, las cuales se han convertido en una nueva herramienta de comunicación. Una red social muy usada en este tipo de casos es Twitter, en donde con tal solo crear una cuenta un usuario puede compartir todo lo que pasa en su alrededor, dar información detallada en tiempo real e inclusive actualizando la información minuto a minuto, a través de *tweets*, los cuales son mensajes cortos que pueden ser incluso hilados por medio de un *hashtag* (un identificador de palabra que lleva antepuesto el símbolo numérico #).

Esta facilidad para que cualquier usuario pueda publicar ha hecho que exista un aumento en la difusión de noticias falsas o “*fake news*”.

En algunas ocasiones, los usuarios no solo difunden noticias falsas, sino información engañosa, esto con el uso de lenguaje figurado¹ como metáforas para expresar sus ideas.

Las metáforas, por sí solas, no deberían afectar, pero, por otro lado, si un usuario de Twitter utiliza una metáfora que trivializa un desastre, podría generar una mayor atención en este tipo de *tweets*.

Un ejemplo de una metáfora, *Un #tsunami de problemas*, donde no nos referimos a un tsunami real, sino a una gran cantidad de situaciones que para alguien podrían sentirse como un tsunami. En este trabajo se exploró el análisis de tuits etiquetados como desastres y no desastres, donde aquellos que no son desastres, son en su mayoría textos con metáfora.

¹ Figurado: Dicho de un sentido: Que no corresponde al literal de una palabra o expresión, pero está relacionado con él por una asociación de ideas. <https://dle.rae.es/figurado>

Tabla 1. Ejemplo de uni-gramas, bi-gramas y tri-gramas con la frase “La niña juega con la pelota”.

Frase: La niña juega con la pelota		
Uni-gramas	Bi-Gramas	Tri-gramas
La	La-niña	La-niña-juega
Niña	niña-juega	niña-juega-con
juega	juega-con	juega-con-la
con	con-la	con-la-pelota
la	la-pelota	
pelota		

Una vez que hayan sido analizados, se pasó a una clasificación automática con diferentes modelos de aprendizaje máquina automático, combinado con los extractores de características SBERT y n-gramas.

2. Marco teórico

2.1. n-gramas

Los n-gramas son subsecuencias de n elementos consecutivos, los cuales pueden contener palabras, números, símbolos y puntuación. El valor de “n” puede ser cualquiera, aunque los más utilizados son los uni-gramas (n=1), bi-gramas (n=2) y tri-gramas (n=3) [3].

Los n-gramas pueden ser utilizados como extractores de características, que sirven como entrada a un modelo, especialmente clasificadores. Este proceso de extracción de características se realiza mediante el conteo de ocurrencia de cada n-grama, donde se calcula la probabilidad condicional y se asigna a una categoría [13].

En el cuadro 1 muestra un ejemplo de uni-gramas, bi-gramas y tri-gramas, para la frase *La niña juega con la pelota*.

2.2. SentenceBERT

SentenceBERT o SBERT es una técnica de procesamiento de lenguaje Natural (PLN), utilizado para codificar oraciones en vectores numéricos, los cuales pueden ser comparados de manera sintáctica y semántica [5]. SBERT utiliza una red neuronal para la codificación de frases, en donde se realiza un mapeo de una oración.

Ya que SBERT utiliza técnicas de transferencia para su adaptación en tareas específicas, puede ser utilizado en modelos tradicionales de aprendizaje automático, es decir, solo extrayendo las características de los datos y convirtiéndolas en vectores. El número de dimensiones base de SBERT, es de 768 dimensiones [5], el cual implicara el número de características extraídas por oración.

2.3. Aprendizaje automático tradicional

El aprendizaje automático o *Machine Learning* (ML), es una rama de la inteligencia artificial o *Artificial Intelligence* (AI), para tareas de clasificación, regresión, reducción dimensional y agrupamiento.

Este tipo de algoritmos son entrenados a partir de un conjunto de datos y generando una salida [6]. Existen un sin fin de algoritmos de aprendizaje máquina tradicional. Para fines de esta investigación nos centramos en los siguientes algoritmos de clasificación:

- **XGBOOST (*Extreme Gradient Boosting*)**: Utiliza la técnica de *boosting*, para mejorar el rendimiento en la predicción. Este algoritmo está enfocado en los llamados “modelos débiles” (árboles de decisión) y realizar una combinación predictiva. XGBOOST, también tiene una técnica de regularización, para evitar un sobre ajuste y mejorar la generalización del modelo[7].
- **KNN (*K-Nearest Neighbors*)**: Este algoritmo se basa en “k” muestras más cercanas a una nueva, todo esto dentro de un espacio de características. Después, el valor objetivo de la nueva muestra se predice en función del promedio de los valores de “k” [8].
- **RF (*Random Forest*)** : Es un algoritmo de aprendizaje supervisado, utilizado para la clasificación y regresión. Este funciona basado la creación de múltiples árboles de decisión y la combinación de sus predicciones individuales. Todo lo anterior, ayuda a evitar el sobre ajuste y mejora la generalización [9].
- **SVM (*Support Vector Machine*)**: Este algoritmo funciona mediante la búsqueda de hiperplanos óptimos, que maximicen el margen de separación entre clases [8].
- **DT (*Decision tree*)**: Funcionan dividiendo los datos en subconjuntos cada vez más pequeños, todo esto a partir de las características de entrada, a. Los nodos internos del árbol, representan las características de entrada y las hojas las decisiones finales [8].

2.4. Metáfora

De acuerdo a la Real Academia Española [10], la metáfora es una figura retórica que consiste en la utilización de una palabra o expresión en un sentido distinto al habitual, estableciendo relación de semejanza entre dos elementos que no son idénticos.

Por ejemplo: “*El sol caía sobre la ciudad como una lluvia de fuego*”, en esta metáfora, se compara la sensación del calor intenso con una lluvia de fuego, lo que da la idea de una fuerza destructiva y peligrosa.

3. Estado del arte

Hoy en día existen diferentes trabajos relacionados con eventos de crisis, a continuación se presentan algunos:

Parrilla-Ferrer et al. [11], hacen uso de la clasificación binaria automática, esto a través de Naive Bayes y SVM. Este trabajo utiliza un conjunto de *tweets* etiquetados a partir de la inundación del 2012 en Habagat, en donde se pueden encontrar *tweets* informativos y no informativos.

Se ha encontrado que el uso de n-gramas ha ayudado a conocer el contexto en algunas tareas de PLN. Thewall Mike et al. [12] encontraron que la combinación de uni-gramas, bi-gramas y tri-gramas proporcionan resultados de hasta un 86.40 % en la métrica de precisión.

Tabla 2. Extracto del conjunto de datos de entrenamiento de Disaster Tweets [18].

id	keyword	location	text	target
48	ablaze	Birmingham	@bbcmdt Wholesale Markets ablaze http://t.co/IHYXEOHY6C	1
49	ablaze	Est. September 2012 - Bristol	We always try to bring the heavy. #metal #RT http://t.co/YA01e0xngw	0
52	ablaze	Philadelphia, PA	Crying out for more! Set me ablaze	0
363	annihilation	United States	Are souls punished annihilation? http://t.co/c1QXJWeQQU http://t.co/Zhp0SOwXRy	0
364	annihilation		@CalFreedomMom @steph93065 not to mention a major contributor to the annihilation of Israel	1
365	annihilation		@willienelson We need help! Horses will die!Please RT &	1
394	annihilation	Chandler, AZ	U.S National Park Services Tonto National Forest: Stop the Annihilation of the Salt River Wild Horse... http://t.co/SB5R7ShcCJ via @Change	1
396	apocalypse	ColoRADO	I'm gonna fight Taylor as soon as I get there.	0
397	apocalypse	sindria	ohH NO FUKURODANI DIDN'T SURVIVE THE APOCALYPSE BOKUTO FEELS HORRIBLE my poor boy my ppor child	1

Así mismo, en esta investigación se encontró que esta técnica de extracción de características tiene buenos resultados con clasificadores como Naive Bayes, SVM, Regresión Logística y Árboles de Decisión.

Los enfoques clásicos no han quedado excluidos, Stowe, K., et al. [13] trabajaron con el conjunto de datos del huracán Sandy del 2012, en el cual recolectaron al rededor de 22.2M de *tweets* con las fechas de 23 de octubre del 2012 al 5 de abril del 2013.

Para el preprocesamiento de datos, utilizaron los enfoques clásicos y para la extracción de características utilizaron uni-gramas y bi-gramas, poniendo especial atención en algunos elementos que generalmente eliminan como los *retweet* y URL. Los resultados más relevantes encontrados con una selección de características fueron de 0.75 %.

Enfocándonos de manera específica en el conjunto de datos *Disaster Tweets*, se observan tres trabajos: El realizado Saji, B., et al. [14] hacen un análisis en el conjunto de datos para la remoción de urls, arrobas, y menciones. Para el preprocesamiento, lematizan, convierten a minúsculas, eliminan palabras de parada y utilizan la codificación *one-hot*.

Para el modelo de clasificación utilizan LSTM(*Long Short Term Memory*), con el cual obtienen el *accuracy* del 95 % en el conjunto de entrenamiento.

El autor autodenominado **wisdomml** en [15] realizan un preprocesamiento similar a Saji, B., et al., a excepción de incluir la remoción de caracteres no alfabéticos y una tokenización, la extracción de características la realizan por medio de TF-IDF (*Term Frequency-Inverse Document Frequency*) combinado con el uso de bi-gramas y tri-gramas.

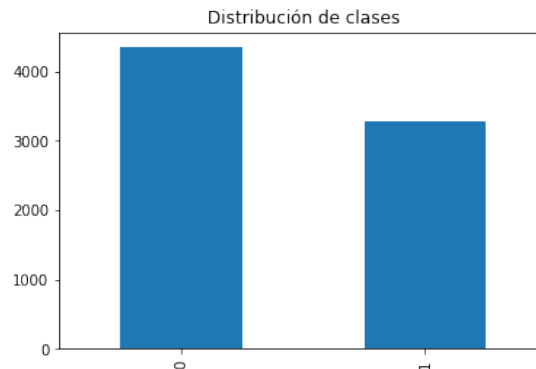


Fig. 1. Distribución de las clases del conjunto de datos *DisasteerTweets*.

Los clasificadores son Naive-Bayes y un *Passive Aggressive Classifier*, estos con validación cruzada. Los mejores resultados obtenidos fueron: Accuracy, bi-gramas y Naive-Bayes con 80.03 %; Precisión, tri-gramas y Naive-Bayes con 86.68 %; F1, tri-gramas con *Passive Aggressive* con 75.18 %.

Por último, en el tercer trabajo realizado por Krishnakumar, M., [16], realizan un preprocesamiento que consiste en eliminar las urls, conversión del texto a minúsculas y la tokenización. Para la clasificación utilizan un *embedding* de tamaño de vocabulario de 10,000 con una longitud máxima de 280, un *max-pooling* de 1 dimensión con tamaño 16, 1 capa densa de 16 y una capa densa final de 1. La métrica reportada es *accuracy*, el cual les da un 77.38 %.

4. Análisis de los datos

4.1. Conjunto de datos

Disaster Tweets [18] es un conjunto de datos de la plataforma Kaggle, donde es resaltado el uso de Twitter como un importante canal de comunicación en tiempos de emergencia.

El conjunto de datos está compuesto por dos archivos de información, entrenamiento y pruebas: Entrenamiento, incluye 5 columnas (id, keyword, location, text y target); Pruebas, incluye 4 columnas (id, keyword, location, text). A continuación, se realizará una descripción del contenido por columna:

- **id:** Identificador de la fila.
- **keyword:** Palabra clave con la que fue etiquetada el desastre natural, el conjunto cuenta con alrededor de 223, por ejemplo: *ablaze*, ardiendo; *sunk*, hundido; *survive*, sobreviviendo; *panic*, pánico; *police*, policía; entre algunos otros. Puede estar vacío.
- **location:** Ubicación del desastre, puede ser una ciudad, país, calle o la localidad (por ejemplo, playa o bosque). Puede estar vacío.
- **text:** Es el texto denominado *tweet*, en donde esta escrita la información.

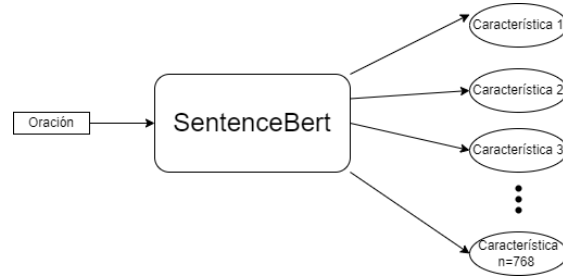


Fig. 4. Ilustración de extracción de características por oración con SBERT.

1. N-gramas,
 2. Sentence Bert.
- Partición de los datos.
 - Clasificadores.
 1. KNN,
 2. Random Forest,
 3. Árboles de decisión,
 4. SVM,
 5. XGBOOST.

5.1. Preprocesamiento

Para obtener los mejores resultados, se realizó un preprocesamiento aprovechando un conjunto de funciones disponibles en un repositorio de GitHub ², además de añadir otros métodos clásicos de preprocesamiento de textos. El tratamiento previo a los datos quedo compuesto de la siguiente manera:

- **Paso de minúsculas a mayúsculas:** Todas las letras pasan a minúsculas.
- **Reemplazo:**
 1. Cantidades numéricas.
 2. Contracciones.
 3. Símbolos matemáticos por descripciones.
- **Remoción:**
 1. Hipervínculos.
 2. Palabras de parada (*stop words*).
 3. Apóstrofes.
 4. Slashes y slashes invertidos.
 5. Alfanuméricos.
 6. Saltos de línea.
 7. ASCII.
 8. Signos de puntuación.

² Macias, C., Soto, M. (2023). Text preprocessing, disponible en <https://github.com/CCogS-Mx/text-preprocessing>

Tabla 3. Resultados de F1, Recall y Precisión con n-gramas.

Método	KNN			Árboles de Decisión			Random Forest		
	F1	Precisión	Recall	F1	Precisión	Recall	F1	Precisión	Recall
1g	0.574	0.788	0.611	0.700	0.759	0.699	0.698	0.816	0.694
1g + 2g	0.523	0.784	0.581	0.704	0.763	0.703	0.608	0.813	0.636
2g	0.501	0.791	0.569	0.541	0.795	0.591	0.503	0.804	0.571
2g + 3g	0.501	0.798	0.569	0.547	0.795	0.595	0.507	0.805	0.573
3g	0.514	0.797	0.576	0.493	0.794	0.575	0.488	0.802	0.562

- 9. Entidades de Twitter.
- 10. Paréntesis.
- 11. Espacios en blanco.
- **Tokenización.**
- **Lematización.**
- **Steaming.**

5.2. Extracción de características:

Se realizaron dos diferentes extracciones de características:

- **Extracción de características mediante N-gramas**
Se utilizó el conteo de vectorización para la extracción de las características basadas en n-gramas. Estas fueron hechas en orden y convertidas en una lista por cada tipo. Se eligieron las siguientes configuraciones de n-gramas: uni-gramas; bi-gramas; tri-gramas; uni-gramas, bi-gramas; bi-gramas, tri-gramas.
- **Extracción de características mediante SentenceBERT**
Para la extracción de características mediante SentenceBERT, se introdujo cada oración dentro del modelo para obtener un vector. Como espacio vectorial se eligieron 768 dimensiones. En la figura 4 se puede visualizar el funcionamiento general del modelo.

5.3. Partición de los datos

El conjunto de datos fue dividido en 80 % para el entrenamiento y en 20 % para validación.

5.4. Modelos

Después de extraer las características, se seleccionaron 5 modelos para la clasificación con la siguiente configuración:

- **XGBOOST:** Fue el clasificador utilizado con la extracción de características de SentenceBERT, incluye una profundidad de 14. Como *booster* utiliza un árbol de gradiente descendente.
- **KNN:** Fue utilizado con la extracción de características de n-gramas, se utilizaron 3 diferentes parámetros de vecinos: 9, 7 y 17.

Tabla 4. Mejores Resultados de F1, Recall y Presicion con n-gramas, para extracción de características de SentenceBERT.

Vector	Métrica		
	F1	Presicion	Recall
Random Forest	0.633	0.651	0.616
XGBOOST	0.625	0.632	0.617
SVC	0.710	0.720	0.710

- **RF:** Este algoritmo fue utilizado en los dos tipos de extracción de características: n-gramas, se utilizaron 200 estimadores y una profundidad de máxima de 32; SBERT, se utilizaron 200 estimadores y profundidad máxima de 50. Esta discordancia en profundidad se debe a la búsqueda de los mejores parámetros en ambos experimentos.
- **SVM:** Este algoritmo fue utilizado para SBERT, incluye un máximo de 1000 iteraciones y un parámetro de mezcla de la red elástica de 0.15.
- **DT:** Este algoritmo solo fue utilizado para la extracción de características de n-gramas, en el cual se implementaron 3 diferentes parámetros de profundidad máxima, 7, 8 y 30.

En algunos casos, se muestra que no se usó la misma extracción de características para todos los clasificadores. Para el caso de los clasificadores que solo usaron la extracción de características de SBERT, como ya se mencionó en la sección 2, esta es una técnica más avanzada que convierte lo extraído a vectores numéricos, en cambio, los n-gramas hacen un conteo de palabras y los vectores numéricos, lo que hace que los vectores numéricos de alta dimensión sean mejor procesados por modelos como XGBoost.

En el caso del clasificador SVM, al ser un clasificador lineal, permite una mejor separación de las clases [19], lo que hace que sea adecuado para trabajar con características extraídas con SBERT.

Por otro lado, las KNN tienen una gran capacidad de manejar características basadas en n-gramas, las cuales son altamente dimensionales [20]. Para el caso de los árboles de decisión, además de tener la misma capacidad de manejar características de esas dimensiones y son altamente interpretables. En adición, fue relevante hacer una revisión del comportamiento de XGBOOST con SBERT y los DT con n-gramas, ya que XGBOOST utiliza de fondo los árboles de decisión.

Para el caso de RF, se estudió una comparación directa del funcionamiento de un algoritmo que es menos propenso al sobre ajuste.

6. Resultados

Se realizaron 15 experimentos para la extracción de características con N-gramas y 4 experimentos para la extracción de características con SBERT. Para la evaluación de estos modelos, se consideró apropiado la utilización de las métricas de F1, precisión y *recall*.

Tabla 5. Comparación respecto al estado del arte.

Comparación	Métricas			
	Accuracy	F1	Precisión	Recall
Saji, B., et al. [14]	95 % * ⁴	-	-	-
wisdomml [15]	80.03 %	75.18 %	86.68 %	-
Krishnakumar, M. [16]	77.38 %	-	-	-
Nuestros resultados	-	70.04 %	81.60 %	70.30 %

La tabla 3 muestra los resultados (resaltados con letras negritas) con la extracción de características n-gramas³.

donde se pueden visualizar las mejores combinaciones y resultados para las métricas propuestas, los cuales son: F1, 1g + 2g con árboles de decisión; Precisión, 1g con Random Forest; Recall 1g + 2g, Árboles de Decisión.

La tabla 4 muestra los resultados con la extracción de características de SBERT. Se puede notar que el mejor clasificador con estas características fue SVC.

Los resultados obtenidos en este análisis pueden ser comparados directamente con los trabajos [14, 15, 16], ya que utilizan el mismo conjunto de datos. Pero, aunque el conjunto sea el mismo, los autores de cada uno de los artículos no se centran en el análisis de los datos contextuales, únicamente en la clasificación de ellos. En la tabla 5 se puede observar esta comparación.

7. Conclusiones y trabajo futuro

En este artículo se presentó el análisis y clasificación automática de *tweets* sobre el conjunto de datos de *DisasterTweets*, a través de algoritmos clásicos de aprendizaje automático y extractores de características como n-gramas y SBERT.

A pesar de que existen múltiples trabajos de clasificación automática con este conjunto de datos e inclusive, resultados superiores, este no fue el único objetivo alcanzado.

Como se observó, los textos contextuales basados en metáforas son extremadamente abstractos y la utilización de una comparación explícita en algunas ocasiones no es clara.

Se observó que el uso de n-gramas combinados es importante para la clasificación, ya que la relación de un texto puede ser capturada por medio de la coocurrencia de los n-gramas, esto se puede verificar al observar que los mejores resultados son cuando existen combinaciones de uni-gramas y bi-gramas.

Por otro lado, aunque SBERT es efectivo y es un extractor muy avanzado, no se toma específicamente para la extracción de los patrones gramaticales o conocer la relación de palabras específicas, y se puede observar en los resultados obtenidos.

Este trabajo nos demuestra que es importante hacer un estudio más profundo en las características principales del texto, es decir, una exploración del conjunto de datos, las uniones, los diferentes patrones, y poner especial “atención”.

³ 1g, unigramas; 2g, bigramas; 3g, trigramas. El símbolo de “+” denota la unión de N-gramas

Como trabajo a futuro se propone aplicar mecanismos de atención, que den mayor peso a las relaciones de las palabras. Específicamente en explorar las capas de atención de BERT (*Bidirectional Encoder Representations from Transformers*) [21], ya que usa un enfoque bidireccional, es decir, va de izquierda a derecha para crear la representación vectorial de alta precisión.

Referencias

1. Naciones Unidas: Las catástrofes relacionadas con el clima se quintuplican en 50 años, pero la mejora de los sistemas de alerta salva más vidas. Noticias ONU (2021) news.un.org/es/story/2021/09/1496142
2. Rodríguez, H., Trainor, J. E.: Emergency response and information exchange during natural disasters: The role of the amateur radio service. *Disasters*, vol. 22, no. 3, pp. 238–250 (1998)
3. Jurafsky, D., Martin, J. H.: *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Education (2019)
4. Manning, C., Schütze, H.: *Foundations of statistical natural language processing*. The MIT Press (1999)
5. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992 (2019) doi: 10.18653/v1/D19-1410
6. Russell, S., Norvig, P.: *Artificial intelligence: A modern approach*. Pearson (2010)
7. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016) doi: 10.1145/2939672.2939785
8. Hasan, M. M., Hossain, M. S., Rahman, M. S.: A comprehensive survey on machine learning techniques. *Applied Sciences*, vol. 11, no. 4 (2021)
9. Nielsen, M.: *Neural networks and deep learning*. Determination Press (2015)
10. Real Academia Española: *Metáfora*. Diccionario de la lengua española (22.^a ed.) (2001) <https://www.rae.es/drae2001/metáfora>
11. Parilla-Ferrer, B. E., Fernández, P. L., Ballena, J. T.: Automatic classification of disaster-related tweets. In: *International conference on Innovative Engineering Technologies*, pp. 62–69 (2015) doi: 10.15242/IIIE.E1214072
12. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418 (2011) doi: 10.1002/asi.21462
13. Stowe, K., Paul, M. J., Palmer, M., Palen, L., Anderson, K. M.: Identifying and categorizing disaster-related tweets. In: *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media*, pp. 1214–1224 (2016) doi: 10.18653/v1/W16-6201
14. Saji, B.: Disaster tweet classification using LSTM - NLP. *Analytics Vidhya* (2022) <https://www.analyticsvidhya.com/blog/2022/05/disaster-tweet-classification-using-lstm-nlp/>
15. Wisdom ML: Disaster tweets classification using machine learning & NLP approach. NLP Project (2022) <https://wisdomml.in/disaster-tweets-classification-using-machine-learning-nlp-approach/>
16. Krishnakumar, M.: Natural language processing with disaster tweets : Part 1. *Medium* (2021) <https://medium.com/@mukilankrishnakumar2002/natural-language-processing-with-disaster-tweets-part-1-db31c9ad07>

17. Liu, J., Singhal, T., Blessing, L., Wood, K., Hui-Lim, K.: CrisisBERT: A robust transformer for crisis classification and contextual crisis embedding. In: Proceedings of the 32nd ACM Conference on Hypertext and Social Media, pp. 133–141 (2021) doi: 10.1145/3465336.3475117
18. Stepanenko, V.: Disaster tweets. Real or not? NLP with disaster tweets challenge add-on. Kaggle (2021) www.kaggle.com/datasets/vstepanenko/disaster-tweets
19. Sun, C., Qui, X., Xu, Y., Huang, X.: How to fine-tune BERT for text classification? (2020) doi.org/10.48550/arXiv.1905.05583
20. Zhang, H., Yu, L., Zhou, M.: A review on recent advances in N-gram language modeling. *Computer Science Review*, vol. 29, pp. 21–39 (2018)
21. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019) doi: 10.18653/v1/N19-1423