

# Análisis de sentimiento en letras basado en la escala valence-arousal

A.V. Rodrigo Ernesto. Universidad del Valle de Guatemala, alv16106@uvg.edu.gt

**Resumen**—La escala valence-arousal busca representar el sentimiento que siente un ser humano en dos ejes. Valencia, que significa que tan positivo o negativo es el sentimiento. Y excitación, que define que tan intensamente se presenta el sentimiento. El presente proyecto busca clasificar, con la ayuda de redes neurales CNN, letras de diferentes canciones populares dentro de esta escala.

## I. INTRODUCCIÓN

El análisis de sentimiento es el proceso de identificar y categorizar opiniones que han sido expresadas en un fragmento de texto, especialmente con la finalidad de determinar si la actitud que toma el autor es positiva o negativa.

Es una de las áreas de procesamiento de lenguaje natural que más ha interesado a los investigadores a lo largo de los años dados sus múltiples usos prácticos. Cualquier negocio puede beneficiarse del análisis de las opiniones de sus clientes para determinar el éxito o fracaso de una campaña de marketing, determinar la calidad del producto o servicio que están prestando, etc.

Una de las áreas en las que el análisis de sentimiento puede ser utilizado es en la industria musical, en la cual su implementación es capaz de producir resultados rentables e interesantes, como recomendadores automáticos de música, detección de estados de ánimo o provisión de feedback a artistas que quieren transmitir emociones específicas a su audiencia.

A lo largo de los años se han utilizado para categorizar emociones escalas basadas en valence (positividad o negatividad) y arousal (intensidad, excitación). Estos dos valores pueden presentarse en dos dimensiones con el fin de crear un plano que representa de manera general las emociones que afectan el ser humano, tal y como se puede observar en la figura 1.

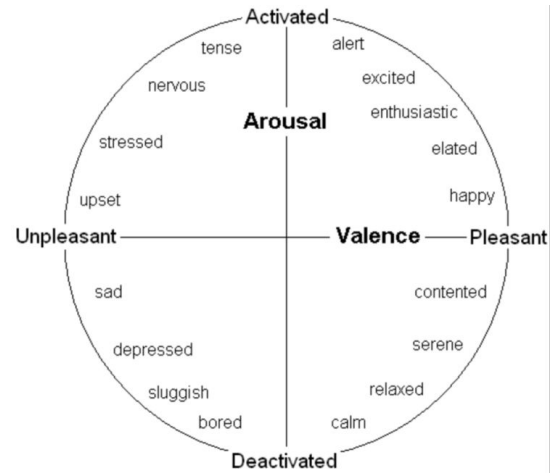


Fig. 1. Plano de Valence-Arousal

## II. TRABAJO RELACIONADO

### A. Text-based Sentiment Analysis and Music Emotion Recognition

Erion Cano investigó la creación y efectos de redes neuronales recurrentes en el análisis de sentimientos para letras [1]. Creo un dataset a partir de tags presentes en LastFM y utilizando word embeddings para agrupar las canciones en categorías que representaban la escala de valence-arousal.

A continuación, utilizó redes neuronales recursivas con la finalidad de clasificar estas canciones en una de sus categorías. Cano consiguió un accuracy de 74%.

## III. APPROACH

El presente proyecto busca analizar, mediante el uso de CNNs y transformers, el texto encontrado en letras de canciones populares respecto a la escala valence-arousal.

Para comenzar, se tomó un dataset conseguido por medio de crowdsourcing que mide sentimientos de una canción en concordancia a la escala valence arousal. Este dataset contiene valores numéricos entre 0 y 1 para ambas medidas. A continuación se separaron las canciones en 4 sectores: relaxed (alto valence, bajo arousal), sad (bajo valence, bajo arousal), happy (alto valence, alto arousal) y angry (bajo valence, alto arousal).

Con este método fue posible obtener alrededor de 2000 datos. Se procedió a descargar las letras de las canciones con ayuda de la herramienta *lyricwikia* y agregarlas al dataset. Este paso redujo la cantidad de datos dado que no fue posible obtener letras para todas las canciones en el dataset.

Se separaron los datos en 3 sets con una distribución de 80-10-10, siendo estos training, validation y test respectivamente. Se consideró aplicar un threshold en las escalas de valence y arousal para eliminar los datos que estuvieran muy “centrados” en la escala y que podían causar

incongruencias y casos límite en nuestro entrenamiento. Lastimosamente, se descarto esa idea dado que el dataset con el que se estaba trabajando ya era bastante pequeño y reducirlo aun mas podria causar mas problemas en el entrenamiento de lo que la eliminación de estos datos ambiguos resuelve.

Las letras fueron limpiadas de caracteres especiales, tokenizadas, y posteriormente truncadas para garantizar uniformidad en el tamaño de los datos.

Se investigaron varios modelos que podrían funcionar para esta tarea de clasificación. Entre ellos, RNNs, CNNs, Single LSTMs, ELMO y Transformers; pero al final se decidió enfocar el proyecto a dos en específico: CNNs y Transformers, dado que CNN es un modelo fácil de entender y prototipar que podía dar un baseline y Transformers es una de las técnicas de NLP que ha creado mejores resultados en los últimos años. Ambas arquitecturas están también disponibles en PyTorch.

#### IV. EXPERIMENTOS

##### A. CNN inicializada de manera random

El primer experimento que se realizó fue una CNN básica con inicialización de pesos random y 3 filtros de tamaño 2, 3 y 4 con max pooling.

En este primer experimento se tomaba en cuenta tanto las letras, como el autor y el título de la canción. Posteriormente tanto el título de la canción como el autor de la misma fueron eliminados dado que no generaban una mejora significativa en el modelo de aprendizaje.

Esta primera iteración demostró ser irritantemente lenta, dado que en esta etapa no se contaba con cuda para agilizar el entrenamiento de la red neural. Además, los word embeddings tardaban muchas épocas para empezar a demostrar mejoras.

Épocas	% training	Loss	% test
100	75%	1.25	35%

Tabla 1. Resultados CNN random

##### B. CNN utilizando word embeddings pre-entrenados y CUDA

Para la segunda iteración se adoptó el uso de CUDA, que redujo el tiempo de entrenamiento de cada época de 15 min a 1 min, permitiéndonos añadir más épocas y probar mas arquitecturas.

El segundo cambio que se realizó fue la utilización de word embeddings pre-entrenados con la finalidad de mejorar la accuracy de nuestro modelo. Se utilizaron 3 diferentes embeddings, pero el que mostró mejores resultados fue *glove*, entrenado con tweets y que contaba con 6B tokens y vectores de 100 dimensiones.

Dados los evidentes problemas de overfitting, se decidió implementar dropout para intentar mitigarlos, lastimosamente ningún valor de dropout parece generar una mejora significativa

Épocas	% training	Loss	% test
200	97.31%	1.551	48.44%

Tabla 1. Resultados CNN Glove y CUDA

##### C. Transformer

El tercer experimento fue la implementación de un transformer, utilizando fine-tuning del modelo BERT. Lastimosamente por restricciones de tiempo y baja cantidad de memoria en la tarjeta gráfica nvidia que se estaba utilizando, no fue posible entrenar el modelo.

#### V. ANÁLISIS

La eliminación de los campos autor y nombre no afectó en el rendimiento de los modelos, esto se debe, en el caso de los autores, a que existen pocas ocurrencias del mismo autor en el dataset, por lo que entrenar para el autor no presenta ninguna mejora para el modelo. En cuanto al título, se había considerado inicialmente que sería algo importante para entrenar el modelo, pero parece ser que su largo tan corto resulta de muy poca ayuda al momento de entrenar los modelos de red neuronal.

Para empezar, la utilización de pre-trained embeddings mejoró sustancialmente tanto la velocidad con la que se conseguían resultados como el accuracy de la red neuronal, lastimosamente los modelos que se generaron tenían un alto grado de overfitting, a pesar del uso de regularización y dropout, no fue posible mejorar el rendimiento de los modelos.

Este problema es potencialmente producto de la baja cantidad de datos con los que se contaba para el entrenamiento de la red neural.

#### VI. CONCLUSIONES

- Una cantidad pequeña de datos genera notables problemas de overfitting.
- Los pre-trained word embeddings mejoran de manera significativa el performance de los modelos de NLP.

#### REFERENCIAS

- [1] E. Cano, "Text-based Sentiment Analysis and Music Emotion Recognition" *Doctoral Program in Computer and Control Engineering, Politecnico di Torino*.
- [2] Yu-Ching Lin, Yi-Hsuan Yang, and Homer H. Chen. Exploiting online music tags for music emotion classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7S(1):26:1–26:16, November 2011
- [3] Çano, Erion; Morisio, Maurizio. Music Mood Dataset Creation Based on Last.fm Tags. In: Fourth International Conference on Artificial Intelligence and Applications, AIAP 2017, Vienna Austria, 27-28 May 2017, pp. 15-26, DOI:10.5121/csit.2017.70603 BibTex