Facultad de Ciencias Naturales y Exactas

Departamento de Ciencia de la Computación

Trabajo de Diploma en Opción al Título de Licenciado en Ciencia de la Computación

Título:
Aprendizaje Profundo para el Perfilado de Usuarios en Redes Sociales

Autor: Roberto Labadie Tamyao

Tutores: Dr. Daniel Castro Castro

M.Sc. Reynier Ortega Bueno

Curso 2020-2021

Contenido

Lista de Figuras

Introducción

Internet se define como la interconexión mundial de redes individuales operadas por el gobierno, la industria, el mundo académico y las partes privadas. En cuestión de muy pocos años, Internet se consolidó como una plataforma muy poderosa que ha cambiado para siempre la forma de comunicarse. Esta manera tan simple y accesible de compartir datos, ha propiciado un desplazamiento de los entes sociales hacia el uso casi exclusivo de este medio.
Un rol fundamental dentro de este universo comunicativo lo juegan las redes sociales donde de acuerdo al sitio DataReportal[1] a finales de 2020 existían más de 3.8 mil millones de usuarios de los 4.5 mil millones de personas conectadas a Internet, lo cual implica un crecimiento desmedido de la cantidad de datos multimodales que se genera a diario.
El mundo del Big Data (Riahi and Riahi, 2018) en el cual nos sumerge este hecho, reporta una situación beneficiosa para el desarrollo de procesos sociales, en cuanto a la cantidad de información brindada por los usuarios . Estos procesos van desde estudios de marketing donde la retroalimentación a partir de opiniones permiten dirigir de manera efectiva la promoción hacia grupos sociales específicos, hasta aplicaciones en el área de la Interacción Humano-Computadora (*Human-Computer Interaction* HCI), donde el conocimiento de las características físicas y psicológicas de las personas permite personalizar la interfaz de comunicación. Sin embargo un cúmulo de información tan significativo, se hace imposible de tratar en un espacio de tiempo razonable a menos que se haga de manera automatizada.
Diversos estudios se han dirigido al diseño de métodos para manejar tal cantidad de información y realizar inferencias a partir de la misma, a la vez que han contribuido al desarrollo en ramas tales como el Procesamiento del Lenguaje Natural (*Natural Langauage Processing* NLP) y Visión por Computadoras (*Computer Vision* CV) en el área de la Inteligencia Artificial (*Artificial Intelligence* AI).
Dentro del NLP, la tarea de Perfilado de Autores (*Author Profiling AP*) (Rosso et al., 2019; Rosso and Rangel Pardo, 2020) se encarga específicamente de realizar un análisis de la información textual elaborada por una persona que permita establecer atributos y patrones de comportamiento para caracterizarla en cuanto sexo, rango de edades o rasgos personales (e.g si la persona es extrovertida o no, ideología política). Sin embargo, para el AP el hecho de que la información recuperada de estos textos varíe enormemente en términos de su formato aun cuando proviene de la misma persona, sumado a que las secuencias texuales constituyen información digital no estructurada, hacen desafiante el proceso de analizarla y clasificarla automáticamente.
Inicialmente este tipo de tareas se desarrollaban sobre contenido generado en textos formales, periódicos, cartas o revistas, sin embargo determinar el perfil de una persona mediante el análisis de su cuenta en una red social ha tomado un gran auge en los últimos años (Rangel et al.,

[1] https://datareportal.com/

2018; Rangel and Rosso, 2019; Cimino et al., 2020). Las redes sociales además de haber logrado acelerar la comunicación entre las personas así como reunir varias formas del pensamiento individual en un mismo espacio, se han convertido en un medio donde se desarrollan procesos negativos tales como, la divulgación de discursos de odio (Rangel et al., 2021), *bullying* o de noticias falsas (Rangel et al., 2020). Este hecho introduce nuevos retos al AP con tareas que involucran la detección de características altamente subjetivas como la ofensa, la toxicidad en el lenguaje y otros patrones psicológicos de comunicación que hacen más complejo el perfilado en relación a otras tareas.

La mayor parte de los avances en el desarrollo de sistemas de AP han sentado sus bases dentro del ambiente académico, reuniendo estudios tanto de la Ciencia de la Computación como de la Lingüística. Algunas de las campañas de evaluación más importantes donde se han compartido tareas de este tipo son *Plagiarism, Authorship and Social Software Misuse* PAN[2] y *Evaluation of NLP and Speech Tools for Italian* EVALITA[3], los que se han dirigido últimamente al análisis del genero textual de micro-blogging con un enfoque multilingüe, empleado en medios sociales como Twitter[4].

Tradicionalmente se han empleado dos tipos de acercamientos que han probado empíricamente ser efectivos para tratar el AP en redes sociales: los *basados en estilo* y los *basados en contenido*. Las propuestas basadas en estilo se refieren al hecho de analizar como los autores se expresan cuando escriben, en cambio la basada en contenido se apoyan en el área temática del texto analizado. La mayor contribución de varios trabajos se ha basado en la selección de atributos que permitan medir el estilo autor y el contenido simultáneamente mediante el empleo de métodos de Aprendizaje de Máquina (*Machine Learning* ML).

Un proceso fundamental en este tipo de propuestas es la selección de rasgos para representar vectorialmente los elementos del texto, este a su vez es un punto crítico en el cual pueden ser eliminados elementos que ayuden al modelo a discernir entre una clase u otra, en el caso de tareas de clasificación, pero también se puede introducir información ruidosa y de igual forma afectar su desempeño.

En los últimos años dentro del Machine Learning ha tenido un auge enorme el Aprendizaje Profundo (*Deep Learning* DL) en las áreas de CV y NLP, estableciendo nuevos estados del arte en la mayoría de sus tareas (Alom et al., 2019). Este auge estuvo condicionado por las capacidades de procesamiento de las nuevas computadoras y la cantidad de datos disponibles. Los modelos de DL han mostrado una gran habilidad para aprender representaciones de rasgos (*features*) con un alto nivel de abstracción. Estas representaciones capturan elementos que son posiblemente omitidos mediante la extracción manual de features y que facilitan el proceso de inferencia de los propios modelos de DL y de los métodos más tradicionales de ML.

Dentro del AP también se ha introducido el Aprendizaje Profundo con resultados alentadores. Sin embargo la mayoría de los modelos alcanzan mejores resultados al analizar la tarea a la que se orienta el perfilado cuando se emplean para clasificar mensajes individuales en lugar del perfil completo, e.g., se desempeñan mejor en la tarea de detección de odio en un tweet que en la detección de un perfil que tiende a usar un discurso de odio.

Por otro lado, las necesidades que cubren las tareas de AP van más allá del idioma en el que la persona postee un mensaje en una red social o escriba una carta, es por ello que urge la

[2]https://pan.webis.de/
[3]http://www.evalita.it/
[4]https://twitter.com/

extensión de los modelos de AI propuestos al esquema multilingüe que predomina en redes sociales como Twitter. Este trabajo de tesis se enmarca en el perfilado de autores en redes sociales empleando técnicas de Aprendizaje Profundo para el modelado y la clasificación de los perfiles teniendo en cuenta el enfoque multilingüe propio de este medio.

Problemática

La mayoría de los trabajos existentes que emplean DL para resolver tareas de perfilado de autores en redes sociales, tanto tradicionales (e.g., determinar rango de edades, sexo, etc.) como las más recientes (e.g. detección de divulgadores de discursos de odio en redes sociales o noticias falsas) ven afectado su desempeño al tratar con las secuencias muy largas que se pueden generar al analizar un perfil completo a la vez, en vez de mensaje a mensaje. La perdida de información (*information vanishing*) en este tipo de secuencias en términos de relaciones a largo plazo que presentan las estructuras sintácticas del lenguaje, es la causa fundamental de esta dificultad. Por esta razón sería conveniente dividir el problema de "clasificar un perfil" en subproblemas y construir una arquitectura modular, en la que primero se modele el perfil y luego este sea clasificado.
Por otro lado en Cuba existe un limitado tratamiento y aplicación de este tipo de métodos automatizados para llevar acabo tareas de impacto como las mencionadas, ya sea en el área forense o empresarial .

Hipótesis

Dividir el proceso de clasificación de un perfil dentro de una tarea en: (1) codificar los mensajes individualmente y (2) modelar-clasificar el perfil, puede ayudar a prevenir la perdida de información generada del análisis de largas secuencias. Esto sumado a la capacidad de representar la información no estructurada de los métodos de DL puede influir positivamente en la precisión de las predicciones. Además, proponer un modelo lo suficientemente robusto, podría incentivar el empleo de esos métodos automatizados de AP en nuestro país.

Objetivos

Objetivo General

Diseñar una arquitectura de Aprendizaje Profundo modular para resolver la tarea de Perfilado de Autores en redes sociales teniendo en cuenta un enfoque multilingüe.

Objetivos Específicos

1. Diseñar una arquitectura que capture rasgos abstractos que permitan clasificar un perfil de usuario de una red social atendiendo tanto a tareas relacionadas con características demográficas de los autores, como con aspectos psicológicos de los mismos.

2. Extender la arquitectura propuesta a un enfoque multilingüe de las tareas evaluadas.

3. Evaluar y analizar el empleo de arquitecturas modulares basadas en DL sobre colecciones de datos propuestas en tareas de AP compartidas en las plataformas PAN (2020, 2021) y EVALITA (2020).

Estructura del Trabajo

Este trabajo esta estructurado en tres capítulos además del introductorio y una última sección en la que se describen las conclusiones de las modelaciones y se proponen caminos a seguir para trabajos futuros. Cada uno de los capítulos se listan a continuación:

- En el Capítulo 1 se describen de manera breve métodos del estado del arte así como conceptos básicos relacionados con el Machine Learning necesarios para la comprensión de este trabajo.
- El Capítulo 2 describe las tareas de AP en las que se evaluarán las arquitecturas propuestas, así como los datos anotados sobre los cuales se basan los procesos de entrenamiento y evaluación.
- En el Capítulo 3 se exponen los experimentos realizados en el proceso de ajuste de cada unos de los modelos y sus módulos sobre los lenguajes estudiados.

.

Capítulo 1

Fundamentos

1.1. Estado del Arte

El Procesamiento del Lenguaje Natural, como cualquier tarea llevada a cabo por un algoritmo de Machine Learning, se basa en el análisis de un conjunto de rasgos del objeto a procesar, con la finalidad de realizar inferencias que permitan al sistema computarizado interactuar con el usuario.
Para la tarea de AP no es diferente, por lo que la mayoría de los trabajos se dirigen al desarrollo de una efectiva selección de rasgos, estableciéndose dos categorías fundamentales; los rasgos de estilo y los de contenido.
El análisis de estilo se enfoca en la detección de rasgos estilísticos del autor que sean lo suficientemente invariantes a lo largo de los pasajes escritos, pero que varíen de un autor a otro. Ejemplos de estos podrían ser la longitud media de las oraciones o la cantidad esperada de símbolos de puntuación, emoticones, preposiciones o adverbios que usa en un párrafo.
Por otra parte el análisis de contenido se enfoca en la información contextual siguiendo la misma estrategia del análisis de estilo, i.e., llevar información estadística de la presencia de determinado conjunto de palabras o estructuras gramaticales. Este tipo de rasgos incluye el uso de n-gramas de palabras y caracteres, *slang words*[1], Bolsas de Palabras (*Bag of Words* BoW) (Pizarro, 2019; Valencia-Valencia et al., 2019), palabras concluyentes (e.g., finalmente, para concluir, en conclusión, etc.) y lexicones de palabras. Dentro de este último método para categorizar palabras, el más empleado es el sistema *Linguistic Inquiry and WordCount LIWC* (Pennebaker et al., 2015) el cual contiene alrededor de 70 diccionarios de lexicones divididos en categorías como; Preocupaciones personales (*Personal Concerns*), Discurso informal (*Informal Speech*), Impulsos (*Drives*) y necesidades fundamentales (*Basics Needs*), etc.

Estos dos grupos de rasgos, de estilo y contenido, son ortogonales puesto que los rasgos que se tienen en cuenta para capturar estilo, son precisamente aquellos que son independientes del tópico, por lo cual se ha hecho habitual el uso de enfoques que los combinen a ambos. Por otro lado el empleo de elementos contextuales implica introducir en el proceso de clasificación un sesgo hacia una clase que este más representada por determinado tema, e.g., según (Schler et al., 2006) estos rasgos pueden facilitar la determinación del sexo del autor ya que por ejemplo los hombres mayormente tienden a hablar de política y noticias, mientras que las mujeres

[1] *slang* se refiere a un lenguaje muy informal empleado por un grupo particular de personas.

se muestran más interesadas por la moda, fiestas y prendas de vestir; sin embargo es posible encontrar una mujer que regularmente postee tweets relacionados con deportes o automóviles, luego este perfil sería potencialmente mal clasificado por un modelo de AP basado en rasgos de contenido.
La mayoría de los trabajos de Machine Learning han usado métodos de clasificación tradicionales como Regresión Logística (Logistic Regression LR) (Valencia-Valencia et al., 2019), Máquina de Vectores de Soporte (*Support Vector Machines* SVM) (Pizarro, 2019) y Bosques Aleatorios (*Random Forest* RF) (Johansson, 2019) combinando conjuntos de rasgos que responden a estas clasificaciones.

Con la introducción del Deep Learning en el NLP esta tendencia a la extracción manual de rasgos de tipo estadístico ha sido desplazada por el aprendizaje de rasgos abstractos que representan a las estructuras gramaticales atendiendo no solamente a su significado semántico como elementos aislados del texto, sino que tienen en cuenta además el contexto en el que son empleados, facilitando la comprensión y clasificación de los textos tanto a los propios modelos de DL como a los tradicionales de ML. Este tipo de enfoques se basan fundamentalmente en el uso de *embeddings* preentrenados con distintas estrategias (Joo and Hwang, 2019; López-Santillán et al., 2019), principalmente Word2Vec (Mikolov et al., 2013), Glove(Pennington et al., 2014) y Fasttext(Bojanowski et al., 2016).
Las Redes Neuronales Artificiales como mecanismos de aprendizaje y clasificación, por su parte han logrado un desempeño superior a los métodos tradicionales de ML en muchas tareas de NLP, teniendo en cuenta que aprenden a decidir que elemento del texto tomar como rasgo significativo a la hora de modelar los objetos. Esquemas especializados en análisis de secuencias, como las Redes Neuronales Recurrentes (RNN) (Dias and Paraboni, 2019; Bakhteev et al., 2020) y las arquitecturas *Transformers* (Transformadoras) (Iyer and Vosoughi, 2020; Baruah et al., 2020) se han empleado satisfactoriamente para el perfilado de autores en los últimos años, pero también se ha extendido el uso de arquitecturas diseñadas inicialmente para el tratamiento de otro tipo de información estructurada como las imágenes, tal es el caso de las Redes Neuronales Convolucionales (CNN) (Petrík and Chudá, 2019; López-Santillán et al., 2019).

En los últimos años dentro de PAN se han propuesto tareas de perfilado que van desde la predicción de sexo y variedad del idioma hasta la detección de perfiles manejados por bots y perfiles que tienden a difundir discursos de odio en el medio social. En la mayoría de estas tareas los trabajos con un mejor desempeño se han enmarcado en técnicas tradicionales de ML, tal es el caso de (Basile et al., 2017) en la tarea *Gender and Language Variety Identification in Twitter at* PAN 2017, quien empleó rasgos a partir de una medida no estándar de frecuencia de términos sobre unigrama de palabras y n-gramas de caracteres para entrenar una SVM y (Martinc et al., 2017) que combinó n-gramas de palabras, caracteres y Elementos del Discurso (*Part of Speech* POS), además de información de sentimientos relacionada con el uso de emojis, conteo de elongación de caracteres, con otros rasgos de estilo para modelar el perfil y entrenar un Regresor Logístico.
La tarea *Bots and Gender Profiling in Twitter at* PAN 2019 consistente en determinar si una cuenta de Twitter pertenece a un bot o a un humano y en el segundo caso, inferir el sexo; (Pizarro, 2019) obtuvo la mejor precisión con una SVM combinando representaciones de los tweets mediante *tf-idf* de n-gramas de palabras y caracteres. De igual forma (Johansson, 2019)

trató de dar solución a la tarea empleando ML con Random Forest y rasgos de estilo como logitud de los tweets, número de letras mayúsculas, URLs, menciones, cantidad de RTs, así como rasgos de contenido, específicamente ocurrencia de términos y etiquetas de POS.
Nuevamente en PAN 2020 en la tarea *Profiling Fake News Spreaders on Twitter*, para detectar divulgadores de noticias falsas, (Pizarro, 2020) se basó en la combinación de vectores de *tf-idf* de n-gramas de palabras y caracteres para representar los tweets y clasificar los perfiles con una SVM. Mientras el modelo propuesto por (Buda and Bolonyai, 2020) consistió en un Regresor Logístico que combina las predicciones de cinco submodelos: (i) n-gramas con Regresor Logístico,(ii) n-gramas con SVM, (iii) n-gramas con Random Forest, (iv) n-gramas con XGBoost y (v) XGBoost con features de estilo.
Diversos modelos de DL empleando las arquitecturas citadas anteriormente (e.g., RNN y CNN) han sido propuestos a lo largo de estas competiciones, aunque ninguno de ellos mostró una precisión superior a la de los métodos tradicionales de ML. Sin embargo en la tarea *Profiling Hate Speech Spreaders on Twitter at* PAN 2021, el modelo con mejor desempeño, propuesto por (Siino et al., 2021) empleó una Red Neuronal Convolucional sobre la representación de los perfiles con *embeddings* de palabras.

Uno de los mayores problemas con los enfoques predominantes es que los rasgos extraídos son muy dependientes del contexto lo cual puede crear una alta sensibilidad de los modelos ante datos que no correspondan a los corpus con los que han sido refinados sus parámetros o como ya hemos expuesto tengan un sesgo hacia alguna clase. Por ejemplo (Newman et al., 2008) expone que las mujeres tienden a usar emoticones con mayor regularidad que los hombres, lo contrario de lo concluido por (Schwartz et al., 2013). Esto nos sugiere que métodos tan rigurosamente refinados como lo son los dependientes de rasgos manualmente extraídos tienen una menor robustez ante arquitecturas *End-to-End* como las de Deep Learing.

1.2. Marco Teórico

En este epígrafe se realiza un acercamiento hacia los temas y arquitecturas de Machine Learning necesarios para la comprensión de los modelos propuestos en el trabajo.

1.2.1. LSTM: Long Short-Term Memory Neural Networks

Las LSTM son un tipo de Redes Neuronales Recurrentes, las cuales están especializadas en el análisis de datos secuenciales. Las RNNs tienen una unidad principal (la unidad recurrente) la cual explora la secuencia de datos de entrada de un elemento a la vez, ya sea de izquierda a derecha o viceversa. Al analizar un elemento para la determinación de su estado oculto, se comparte la información capturada en pasos anteriores del recorriendo. Esto es, sea h_{t-1} el último estado oculto computado, $x_t \in \Re^d$ el $t - esimo$ elemento de la secuencia de entrada y f una función de no linealidad. El estado oculto actual, se define como:

$$h_t = f(W_x x_t + W_h h_{t-1} + b_h) \tag{1.1}$$

Donde $W_x \in Re^{n_u \times d}$ y $W_h \in Re^{n_u \times n_u}$ son matrices de parámetros y $b_h \in R^{n_u}$ el término de sesgo (*bias*), con n_u el número de neuronas y d la dimensión de los vectores que representan a los elementos de la secuencia. De esta forma la arquitectura aprende a considerar la información

que tiene determinada influencia sobre el elemento de la secuencia que se analiza en cada paso como se muestra en la Figura 1.1, lo que le otorga una especie de "memoria".

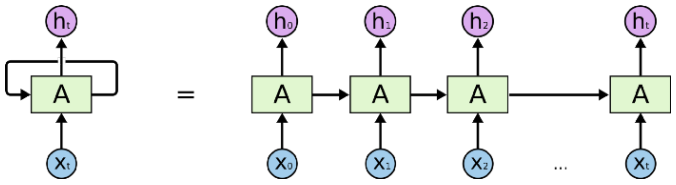


Figura 1.1: Red Neuronal Recurrente sobre la secuencia X_t. (Agarwala et al., 2017)

Sin embargo, debido a que en las RNNs durante el proceso de *backpropagation*, en el que se ajustan los parámetros de la red, cada neurona en la unidad principal calcula su gradiente para un paso del recorrido de la secuencia con respecto a su estado en el paso posterior, mediante la ley de la cadena, ocurre un decrecimiento exponencial de los valores de las derivadas parciales conocido como *gradient vanishing*, lo que hace que los parámetros a penas se actualicen y se dificulte el aprendizaje de relaciones a largo plazo, de aquí su "Short-Term Memory".
Esta limitación es lo que las LSTM tratan de solucionar introduciendo compuertas que deciden que información preservar u "olvidar" de los estados previos en el recorrido por la secuencia de la siguiente forma:
Sean $W_f, W_i, W_o \in \Re^{n_u \times d}$ y $U_f, U_i, U_o \in \Re^{n_u \times n_u}$ las matrices de parámetros de la compuerta de "olvidar ", entrada y salida respectivamente y $b_f, b_i, b_o \in \Re^{n_u}$ sus respectivos términos de bias:

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \end{aligned} \tag{1.2}$$

Una codificación potencial $\hat{c}_t$ considerando el elemento x_t de la secuencia y el estado previo h_{t-1} esta dado por:

$$\hat{c}_t = \sigma(W_c x_t + U_c h_{t-1} + b_c) \tag{1.3}$$

Donde $W_c \in \Re^{n_u \times d}$, $U_c \in \Re^{n_u \times n_u}$ y $b_c \in \Re^{n_u}$. Luego la codificación x_t teniendo en cuenta la codificación del elemento anterior y h_t quedan definidos por:

$$\begin{aligned} c_t &= f_t c_{t-1} + i_t \tanh(\hat{c}_t) \\ h_t &= o_t \tanh(c_t) \end{aligned} \tag{1.4}$$

1.2.2. Redes Neuronales Convolucionales sobre secuencias

La arquitectura de una CNN (LeCun et al., 1998) sobre una secuencia de texto es capaz de capturar dependencias temporales a corto plazo mediante filtros unidimensional que analizan n-gramas de palabras o caracteres y cuyos parámetros son compartidos durante cada paso de manera similar a las LSTM.
Esto es, sea $X \in \Re^{l \times d}$ la secuencia de entrada de longitud l donde cada elemento es un vector $d-dimensional$, F_k el conjunto de filtros con ventana k de una capa convolucional, cada uno de los $F_{ki} \in \Re^{k \times d}$ son inicializados de manera independiente y de la misma forma aprenden a

capturar relaciones a corto plazo dentro en X. La operación convolución (*conv-op*) transforma a X en una nueva secuencia $X' \in \Re^{(l-k)\times n_f}$ donde $n_f = |F_k|$ de la siguiente forma:

$$X'_{ij} = \sum X_{[i:i+k]} * F_{kj} \quad para \quad j \in [0, F_k - 1],\ i \in [0, l-k] \tag{1.5}$$

Como se puede observar en la Figura 1.2:

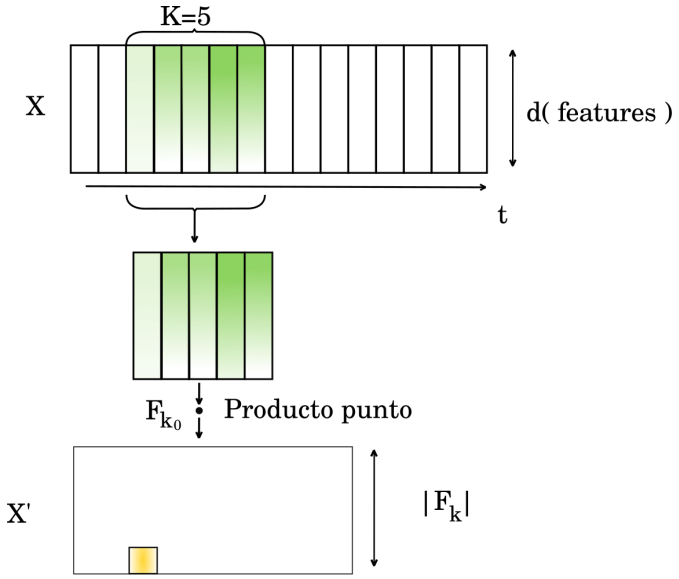


Figura 1.2: Operación convolución sobre X con filtros de tamaño de ventana k.

Vinculada a la *conv-op* en este tipo de arquitectura se suele emplear además una capa de *pooling*, en la cual se combinan los valores de elementos en una vecindad para reducir la dimensionalidad de la secuencia o se seleccionan los valores más importantes de dicha vecindad, esta operación de *pooling* con una ventana de tamaño k sobre una secuencia X esta definida por:

$$X'_i = f(X_{[i:i+k]}) \ \ para \quad i \in [0, l-k] \tag{1.6}$$

Donde f es la función de *pooling*, regularmente empleadas las funciones Max o Avg.

1.2.3. Mecanismos de Atención

Los mecanismos de atención, son técnicas de procesamiento de las entradas de una arquitectura o de algún resultado intermedio que permiten a la red prestar mas "atención" a elementos específicos de una secuencia o establecer la importancia relativa sobre un elemento del resto. En la práctica, la atención permite a las redes neuronales aproximarse al mecanismo de atención visual que utilizan los humanos.

Self-Attention

Sea x_t la representación del $t-esimo$ elemento de la secuencia, una capa de *self-attention* (auto-atención), captura en una matriz A cuan similar es x_t con sus vecinos. Específicamente

$\alpha_{t,t'} \in A$ expresa la relación de x_t con x'_t y de manera similar al de una red recurrente este valor se calcula como:

$$\begin{aligned} g_{t,t'} &= \tanh(W_x x_t + W_{xt'} x_{t'} + b_g) \\ a_{t,t'} &= \sigma(W_a g_{t,t'} + b_a) \end{aligned} \tag{1.7}$$

Donde σ es la función sigmoide, $W_x, W_{xt'} \in \Re^{n_u \times d}$ son las matrices de parámetros encargadas de codificar la información de x y x' para expresar su compatibilidad, $W_a \in \Re^{n_u \times n_u}$ la matriz de parámetros correspondiente a su combinación no lineal y b_g y b_a los correspondientes términos de bias.
A partir de A el estado correspondiente al vector x_t, $\hat{x}_t$ está dado por la suma ponderada de sus elementos vecinos x'_t:

$$\hat{x}_t = \sum_{i=0} a_{t,t'} x_{t'} \tag{1.8}$$

Luego $\hat{x}_t$ expresa cuan atendido debe ser x_t condicionado por el contexto de su vecindad.

Scaled Dot-Product Attention

El mecanismo *Scaled Dot-Product Attention* (Atención con Producto-Punto Escalado) primero se mapea cada elemento de la secuencia con tres representaciones (*query* y un par *key-value*) para calcular el índice de compatibilidad entre cada par de elementos. Luego, para cada x_t es evaluada su compatibilidad con respecto a cada uno de los elementos vecinos relacionando su *query* (q_t) con las *keys* de los vecinos ($k_{t'}$), estos valores de compatibilidad son escalados y normalizados con la función $softmax$ y empleados para ponderar los vectores de *value* (v_t). Finalmente la representación de $\hat{x}_t$ es calculada como la suma ponderada de los v_t. En forma matricial queda expresado como:

$$Attention(Q, V, K) = softmax(\frac{Q \times K^T}{\sqrt{d_k}}) \times V \tag{1.9}$$

Donde $Q, K \in \Re^{n \times d_k}$ y $V \in \Re^{n \times d_v}$ son el resultado del producto de las matrices de parámetros de *query*, *key* y *value* con los elementos de la secuencia respectivamente, por tanto en la fila t contienen el mapeo del elemento x_t y d_k, d_v corresponden a la dimensionalidad de las vectores de *key* y *value* respectivamente.

1.2.4. Transformers

La arquitectura Transformer (Figura 1.3) basada en mecanismos de atención, específicamente *multihead attention* (Vaswani et al., 2017), está diseñada para tratar problemas de *machine translation* y la conforman dos módulos, el primero conocido como *Encoder* (Codificador) es alimentado con una secuencia textual y se encarga de encontrar una codificación para cada elemento teniendo en cuenta la información de su contexto. El segundo módulo, *Decoder* produce los elementos de una nueva secuencia en el modelado de lenguaje, haciéndolo de uno a la vez y teniendo en cuenta los elementos generados anteriormente y las codificaciones obtenidas por el Encoder de la secuencia de entrada.
La principal ventaja de las Transformers con respecto a las arquitecturas secuenciales más tradicionales, e.g., GRU y LSTM es que en vez de analizar la información textual en una

dirección, esta toma en consideración la entrada completa relacionando cada elemento con el contexto de su vecindad simultáneamente lo cual evita el problema de "memoria" a corto plazo de las RNN. Sin embargo pudiera parecer que existe una pérdida de la percepción del tiempo al analizarlo todo a la vez, es por esto que ademas de representar el texto con un *embedding* de palabras, se tiene en cuenta un *encoding* de posición.

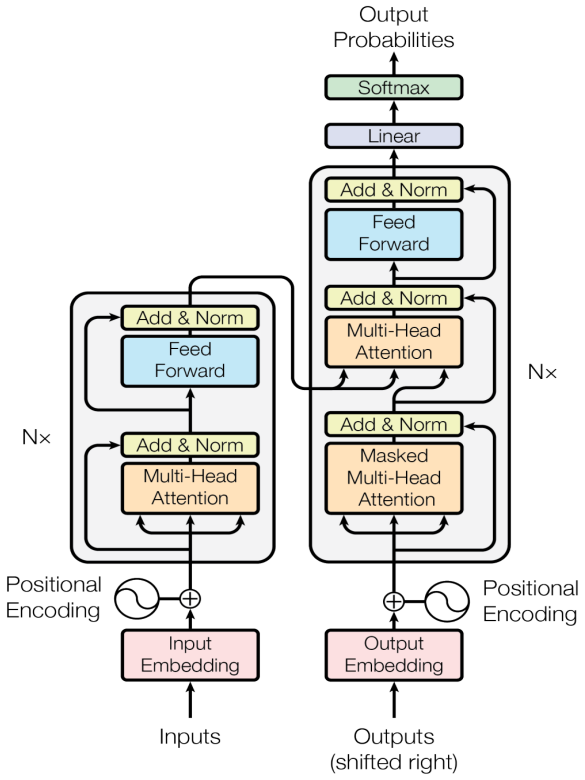
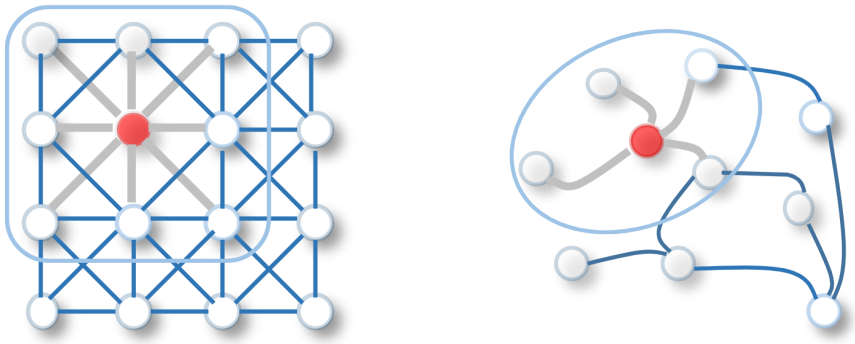


Figura 1.3: Arquitectura Transformer. Módulo Izquierdo, Codificador. Módulo Derecho, Decodificador. (Vaswani et al., 2017)

En la Figura 1.3 se observan cada uno de los módulos los cuales se apilan (N_x), i.e., antes de que el Decoder reciba la informacion codificada de la secuencia de entrada, esta transita por una serie de bloques codificadores haciendo uso de una red residual para prevenir cualquier pérdida de la información.
Dentro de la mayoría de las tareas de NLP este tipo de modelo ha alcanzado nuevos estados del arte simplemente haciendo *transfer learning* del modulo de Encoder hacia la tarea específica. Este módulo es entrenado en dos tareas, (i) predecir palabras enmascaradas de la secuencia, (ii) dadas dos oraciones, predecir si una está a continuación de la otra en el texto, lo cual parece hacer que el modelo llegue a "entender" el funcionamiento del lenguaje y sea relativamente fácil de entrenar sobre otras tareas como el análisis de sentimientos.

1.2.5. Redes Neuronales Convolucionales en Grafos

La principal diferencia entre las CNNs y las Redes Neuronales Convolucionales en Grafos (*Graph Convolutional Neural Nets* GNN) es que las primeras están diseñadas especialmente para tratar datos regularmente estructurados (Euclideanos), mientras que las GNN son su versión generalizada capaz de extraer información en datos donde no existe una relación de orden entre un nodo y otro y las conexiones entre cada uno de ellos es variable (datos irregulares o datos estructurados no Euclideanos), estas dos propiedades son totalmente opuestas a las relaciones que existen entre los píxeles de una imagen (Figura 1.4).



(a) Datos Estructurados Euclideanos

(b) Datos con Estructura No Euclidea-na

Figura 1.4: Relación de Estructura en los Datos. (Wu et al., 2021)

Un uso típico de las GNN es la clasificación de nodos en una estructura. En este tipo de problema el nodo v se caracteriza por un conjunto de features x_v y corresponde a una clase t_v. Dado un grafo $G = (V, E)$ parcialmente etiquetado, nuestro objetivo es predecir a que clase corresponde cada nodo no etiquetado. Mediante una GNN, cada nodo comparte información con su vecindad y transforma su representación a un estado h_v de manera que exprese como este pertenece a su contexto. Específicamente,

$$h_v = f(x_v, E_v, H_v, X_v) \tag{1.10}$$

Donde $E_v = \{(i,j)|(i,j) \in E, v \in \{i,j\}\}$, $H_v = \{h_u|u \in \mathcal{N}(v)\}$ con $\mathcal{N}(v)$ el conjunto de nodos de la vecindad de v y $X_v = \{x_u|u \in \mathcal{N}(v)\}$. Dentro del espacio imagen de f es de nuestro interés que cada nodo tenga un estado h_v unívoco, por lo que apoyándose en el teorema del punto fijo (Brown et al., 1988) es posible encontrar en un proceso iterativo de actualizaciones con determinados parámetros para nuestra función f este h_v contextualizado. El proceso de actualizaciones llevado a cabo dado f es conocido como paso de mensajes. Luego, en forma matricial el estado de todos los nodos en la actualización $t+1$ esta dado por:

$$H^{t+1} = F(H^t, X) \tag{1.11}$$

Con H y X matrices donde a cada fila i le corresponde h_i y x_i respectivamente. Luego, la clasificación de un nodo es llevada a cabo mediante la una nueva función g:

$$t_v = g(h_v, x_v) \tag{1.12}$$

Muchos diseños de la función de agregación han sido estudiados (Kipf and Welling, 2017) teniendo en cuenta propiedades de la topología del grafo y la simetría que debe cumplir el análisis de $\mathcal{N}$ i.e., F no puede ser sensible al orden de agregación de la información. En este trabajo, hacemos uso específicamente de las GNN de tipo espectral (Wu et al., 2021) y la clasificación de G. Este tipo de enfoque no difiere mucho de la clasificación de nodos, pues sigue siendo fundamental la determinación de un estado para cada nodo que exprese su información contextual. Esta información contextual es agregada mediante una operación de *pooling* para alimentar una red densa y proceder con la clasificación de la estructura.

1.2.6. Information Gain

El índice de *Information Gain* IG (Captura de Información) (Koller and Sahami, 1996; Sebastiani, 2002) mide cuanta información aporta un rasgo sobre una clase tomando en cuenta tanto la presencia como la ausencia del término en los documentos pertenecientes a la misma. El IG de un término t en una clase C está definido por:

$$IG(t, C) = \sum_{c \in \{C, \bar{C}\}} \sum_{x \in \{t, \bar{t}\}} P(x, c) \log_2 \frac{P(x, c)}{P(x)P(c)} \tag{1.13}$$

Donde las probabilidades están interpretadas en un espacio de eventos sobre los documentos, e.g., $P(\bar{t}, C)$ indica la probabilidad de que para un documento aleatorio d, el término t no ocurra en d y d pertenezca a la categoría C.

Capítulo 2

Framework

El esquema común para clasificar perfiles teniendo en cuenta cierta característica consiste en: i) extraer rasgos textuales de los documentos del autor, en nuestro caso de los tweets; ii) construir una representación a nivel de tweet o del perfil y finalmente iii) entrenar un modelo de clasificación a partir de la representación elaborada.
Como se puede observar, existe un proceso de identificación de los features empleados para contrastar las características entre tipos de perfiles, haciendo que el desempeño del modelo de clasificación dependa directamente de la robustez de este paso.
Mediante la explotación de modelos de Aprendizaje Profundo, se pretende prescindir de rasgos extraídos manualmente que pudieran resultar ruidosos a los modelos clasificadores o dejar de capturar relaciones claves de la estructura semántica y/o sintáctica, como se expone en la Sección 1.1.

En este Capítulo se describe un *framework* para llevar a cabo el perfilado de autores, basado en el empleo de rasgos abstractos obtenidos a partir de diferentes arquitecturas de DL y la combinación de las mismas. De manera general sus principales contribuciones son:

- Se propone un framework para el perfilado semi-supervisado de autores en redes sociales. El mismo lleva a cabo el aprendizaje de rasgos abstractos para la representación en un espacio latente de los tweets, y construye a partir de los mismos el modelado del perfil.
- Es presenta una combinación de rasgos de estilo a niveles de palabra, oración y otras estructuras gramaticales, para evaluar su influencia sobre la representación exclusiva mediante features extraidos por los modelos de DL.
- Se exponen distintas arquitecturas para modelar tweets y perfiles de usuarios dentro del mismo framework, que relacionen de manera diferente la información para realizar un estudio comparativo.

2.1. Representación de Rasgos a Nivel de Tweet

El análisis del historial de un perfil de usuario en redes sociales como Twitter, involucra una cantidad considerable de información textual, lo cual puede resultar desafiante para el entrenamiento de algunos modelos de DL, sobretodo debido a las relaciones a largo plazo que

es necesario contemplar a la hora de clasificar un perfil o extraer sus rasgos. Este fenómeno es independiente del nivel de supervisado con que se afronte la tarea y es conocido como *information vanishing* (Hochreiter et al., 2001). Otros modelos son capaces de lidiar con este problema, sin embargo, están limitados por la complejidad temporal que poseen para analizar secuencias de texto, tal es el caso de la popular arquitectura *transformer* donde cada capa de atención tiene $O(n^2 * d)$, donde n y d son la longitud de la secuencia y la dimensionalidad de sus elementos respectivamente.
El enfoque empleado para este trabajo se apoya en una arquitectura modular para modelar la información textual primeramente a nivel de tweets.

2.1.1. CNN + Attention + LSTM

Referencias Bibliográficas

Nipun Agarwala, Yuki Inoue, and Alex Sly. 2017. Music composition using recurrent neural networks. *CS 224n: Natural Language Processing with Deep Learning, Spring*.

Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Mahmudul Hasan, Brian C. Van Essen, Abdul A. S. Awwal, and Vijayan K. Asari. 2019. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8(3).

Oleg Bakhteev, Aleksandr Ogaltsov, and Petr Ostroukhov. 2020. Fake News Spreader Detection Using Neural Tweet Aggregation—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Arup Baruah, Kaushik Amar Das, Ferdous Ahmed Barbhuiya, and Kuntal Dey. 2020. Automatic Detection of Fake News Spreaders Using BERT—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. 2017. N-GrAM: New Groningen Author-profiling Model—Notebook for PAN at CLEF 2017. In *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland*. CEUR-WS.org.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

R.F. Brown, Calif.) Conference on Fixed Point Theory (1986, Berkeley, American Mathematical Society, and Calif.) International Congress of Mathematicians (1986, Berkeley. 1988. *Fixed Point Theory and Its Applications: Proceedings of a Conference Held at the International Congress of Mathematicians, August 4-6, 1986*. Contemporary mathematics - American Mathematical Society. American Mathematical Society.

Jakab Buda and Flora Bolonyai. 2020. An Ensemble Model Using N-grams and Statistical Featuresto Identify Fake News Spreaders on Twitter—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Andrea Cimino, Felice Dell'Orletta, and Malvina Nissim. 2020. TAG-it@ EVALITA 2020: Overview of the Topic, Age, and Gender Prediction Task for Italian. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020), Online. CEUR. org*. CEUR Workshop Proceedings (CEUR-WS.org). Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, EVALITA 2020 ; Conference date: 17-12-2020.

Rafael Felipe Sandroni Dias and Ivandré Paraboni. 2019. Combined CNN+RNN bot and gender profiling. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Aarish Iyer and Soroush Vosoughi. 2020. Style Change Detection Using BERT—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Fredrik Johansson. 2019. Supervised Classification of Twitter Accounts Based on Textual Content of Tweets. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Youngjun Joo and Inchon Hwang. 2019. Author Profiling on Social Media: An Ensemble Learning Approach using Various Features. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Daphne Koller and Mehran Sahami. 1996. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 284–292, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Jesus López-Santillán, Luis Carlos González-Gurrola, Manuel Montes-y-Gómez, Graciela Ramírez Alonso, and Olanda Prieto-Ordaz. 2019. An Evolutionary Approach to Build User Representations for Profiling of Bots and Humans in Twitter. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Matej Martinc, Iza Škrjanec, Katja Zupan, and Senja Pollak. 2017. PAN 2017: Author Profiling - Gender and Language Variety Prediction—Notebook for PAN at CLEF 2017. In *CLEF 2017 Evaluation Labs and Workshop – Working Notes Papers, 11-14 September, Dublin, Ireland*. CEUR-WS.org.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

M. Newman, Carla J. Groom, Lori D. Handelman, and J. Pennebaker. 2008. Gender differences in language use: An analysis of 14,000 text samples. *Discourse Processes*, 45:211 – 236.

James W. Pennebaker, Ryan Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. University of Texas at Austin.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Juraj Petrík and Daniela Chudá. 2019. Bots and gender profiling with convolutional hierarchical recurrent neural network. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Juan Pizarro. 2019. Using N-grams to detect Bots on Twitter. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Juan Pizarro. 2020. Using N-grams to detect Fake News Spreaders on Twitter—Notebook for PAN at CLEF 2020. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Francisco Rangel, Anastasia Giachanou, Bilal Ghanem, and Paolo Rosso. 2020. Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Francisco Rangel, Manuel Montes-y-Gómez, Martin Potthast, and Benno Stein. 2018. Overview of the 6th Author Profiling Task at PAN 2018: Cross-domain Authorship Attribution and Style Change Detection. In *CLEF 2018 Evaluation Labs and Workshop – Working Notes Papers, 10-14 September, Avignon, France*. CEUR-WS.org.

Francisco Rangel and Paolo Rosso. 2019. Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Francisco Rangel, GLDLP Sarracén, BERTa Chulvi, Elisabetta Fersini, and Paolo Rosso. 2021. Profiling hate speech spreaders on twitter task at pan 2021. In *CLEF*.

Youssra Riahi and S. Riahi. 2018. Big Data and Big Data Analytics: Concepts, Types and Technologies. *International Journal of Research and Engineering*, 5:524–528.

Paolo Rosso, Martin Potthast, Benno Stein, Efstathios Stamatatos, Francisco Rangel, and Walter Daelemans. 2019. Evolution of the pan lab on digital text forensics. In Nicola Ferro and Carol Peters, editors, *Information Retrieval Evaluation in a Changing World: Lessons Learned from 20 Years of CLEF*, pages 461–485. Springer International Publishing, Cham.

Paolo Rosso and Francisco Rangel Pardo. 2020. Author Profiling Tracks at FIRE. *FIRE 10th Anniversary, SN Computer Science*, 1.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

H. A. Schwartz, J. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie Ramones, Megha Agrawal, Achal Shah, M. Kosinski, D. Stillwell, M. Seligman, and L. Ungar. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLoS ONE*, 8.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Marco Siino, Elisa Di Nuovo, Ilenia Tinnirello, and Marco La Cascia. 2021. Detection of Hate Speech Spreaders using Convolutional Neural Networks. In *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Alex I. Valencia-Valencia, Helena Gómez-Adorno, Christopher Stephens Rhodes, and Gibran Fuentes Pineda. 2019. Bots and Gender Identification Based on Stylometry of Tweet Minimal Structure and n-grams Model. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24.