

Atribución de autoría en textos en español a partir de sus atributos textuales

Fernando Hernández-Ibarra¹, Belém Priego-Sánchez¹, David Pinto²

¹ Universidad Autónoma Metropolitana,
Departamento de Sistemas,
México

² Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{herr.taquitos, belemps}@gmail.com, dpinto@cs.buap.mx

Resumen. La atribución de autoría busca identificar al autor de entre un grupo de posibles autores de un texto. Las aplicaciones de esta tarea abarcan por ejemplo la detección de plagio, identificación del autor de un texto anónimo y análisis forense. En este trabajo se pretende resolver esta tarea para dos corpora, uno compuesto por textos literarios de 9 autores hispanohablantes, y el segundo es un corpus obtenido de la competencia PAN 2017 que se compone de tweets en español, con base en los atributos textuales. Para el primer corpus, se alcanza una precisión sobre un conjunto de pruebas de 89.59% mediante redes neuronales basadas en una arquitectura tipo transformer. Para el segundo corpus, el mejor resultado obtenido fue con una red neuronal convolucional con una precisión de 73.10%. Los resultados obtenidos, en ambos corpora, son prometedores para la tarea de la atribución de autoría.

Palabras clave: Atribución de autoría, aprendizaje profundo, textos en español.

Authorship Attribution in Spanish Text through Text Attributes

Abstract. Authorship attribution seeks to identify the author of a text between a set of possible authors. It can be applied to plagiarism detection, author identification in anonymous text or forensic analysis, and others. This article aims to solve this task through text attributes on two corpora: one composed of literary texts written by 9 Spanish-speaking authors and the second corpus is obtained from the competition PAN 2017, which is composed of tweets in Spanish. The accuracy achieved on the first corpus was 89.59% on a test dataset using neural

networks based on a transformer type architecture. For the second corpus, the best result achieved was an accuracy of 73.10% using a convolutional neural network. The results obtained, on both corpora, are promising for the authorship attribution task.

Keywords: Authorship attribution, deep learning, Spanish text.

1. Introducción

Cada escrito tiene características que los hacen o no similares entre sí más allá del tema que traten; por ejemplo, la riqueza de vocabulario, longitud de oraciones, puntuación, entre otras características que en conjunto forman patrones de redacción propios de cada individuo. Estas marcas personales, que pueden ser una decisión consciente o no, pueden ser cuantificadas y, por tanto, se pueden convertir en una huella lingüística de su autor.

La estilometría es un campo que se ocupa del estudio de estos patrones lingüísticos y es la base para la tarea de identificación de autor, o atribución de autoría (AA), que consiste en, dado un texto, determinar al autor del texto de entre un grupo de posibles autores y el perfilado de autor, que busca descubrir características asociadas al autor tales como el género, la edad o el idioma (incluyendo variantes de éste).

Estas tareas tienen varias aplicaciones: análisis forense, detección de plagio, seguridad e incluso marketing. En el área del Procesamiento del Lenguaje Natural (PLN) la AA se ha vuelto un tema habitual y muchos trabajos se acercan justamente mediante la estilometría, aunque las características más útiles para llevar a cabo la tarea no son fijas para todo estudio, sino que cada conjunto de datos es mejor representado por unas u otras características.

El aprendizaje automático clásico ha sido ampliamente usado, sin embargo, en años recientes y aprovechando la enorme cantidad de datos disponibles en línea, modelos neuronales han ido ganando atención para diversas tareas relacionadas al PLN, en particular, aunque no limitado a, redes neuronales convolucionales (RNC), que son capaces de trabajar con datos donde existan patrones espaciales, redes neuronales recurrentes (RNR) que están diseñadas para trabajar con datos secuenciales o *transformers*, que emplean un mecanismo que permite superar los límites de las entonces dominantes RNR.

Además, es importante recalcar que el aprendizaje profundo no reemplaza al aprendizaje automático clásico, existen trabajos sobre AA donde emplean tanto algoritmos de DL como de ML, obteniendo resultados similares para ambos enfoques o donde uno sobresale al otro de acuerdo a ciertas condiciones de los modelos, como en [1], donde utilizan algoritmos clásicos de aprendizaje automático y una RNR, concluyendo que, bajo la configuración hecha de la RNR, ésta no destaca de forma sustancial sobre sus contrapartes, o en [2] donde comparan un modelo de red neuronal recursiva contra un modelo multinomial Naive Bayes, siendo el modelo de red neuronal mejor cuando se trata de resolver atribución de autoría para 3 autores, pero el modelo Naive Bayes obtiene mejores resultados cuando se escala el problema a 10 autores.

En este trabajo se busca resolver la AA para un corpus compuesto por textos literarios de 9 autores hispanohablantes y para un segundo corpus compuesto de tweets en español, el cual fue tomado de la competencia PAN del 2017 [3], por medio de las características textuales y empleando algoritmos de aprendizaje profundo.

En la sección 2 se presentan algunos trabajos relacionados a la AA resueltos por medio de diferentes enfoques, tanto del análisis de los datos como de los algoritmos empleados. En la sección 3 se describen con más detalle los conjuntos de datos utilizados en el presente trabajo. La sección 4 detalla los pasos que componen la metodología propuesta para la resolución de AA para los dos corpora utilizados.

En la sección 5 se presentan los mejores resultados obtenidos para cada corpus empleando una métrica de evaluación extra a la precisión cruda y se discuten los mismos. Finalmente, la sección expone las conclusiones y posibles mejoras al presente trabajo.

2. Trabajos relacionados

En [1] trabajan la AA utilizando tres enfoques: el primero realiza un análisis a nivel de artículo y utilizan algoritmos clásicos de aprendizaje automático, el segundo se acerca por medio de un análisis a nivel de palabra entrenando un modelo GloVe, y en el tercer enfoque desarrollan una RNR con vectores de palabras pre-entrenados con GloVe y el análisis es a nivel de oración.

Al final del trabajo comparan los resultados de los diferentes algoritmos utilizados, donde se observa que la RNR no destaca tanto del resto de algoritmos, aunque queda abierto a una optimización de la red con la cuál pueda ser que sí se logre un resultado mucho más destacable sobre los enfoques de aprendizaje automático clásico. En [2] buscan un modelo de aprendizaje profundo para resolver el problema de atribución de autoría múltiple, es decir, cuando son varios los autores de un texto, a nivel de oraciones.

Utilizan un corpus compuesto por artículos de Wikipedia que fueron escritos por varios autores. Los resultados demuestran que el modelo de red neuronal recursiva es superado por un modelo Multinomial Naive Bayes conforme se incrementa el número de autores a distinguir. En [4] se aborda la tarea de atribución de autoría haciendo uso de un modelo de ensamble de redes neuronales convolucionales y recurrentes LSTM.

Utiliza un corpus con textos en inglés obtenidos del proyecto Gutenberg, además de los corpus usados en la competencia PAN del 2013 [5] y 2014 [6] para comprobar la generalidad del modelo. Se trabajó dividiendo los textos en n-gramas a nivel de palabra y caracteres, sin aplicar ninguna técnica de tokenización. En [7] se propone un modelo para resolver la tarea de atribución de autoría no supervisada.

Utiliza el mismo corpus que fue dado en el PAN 2017 para la tarea de agrupación de documentos por autor, que incluye dos géneros (artículos y reseñas) y tres idiomas (inglés, holandés y griego). Con base en el estado del arte para esta tarea, propone modificaciones (como el tratamiento que reciben los tokens, el uso de caracteres especiales como puntuación y selección de características), que llevan a una ligera mejora en los resultados mostrados por el estado del arte.

En [8] proponen una solución al problema de atribución de autoría de un solo autor mediante modelos de cálculo de semejanza entre textos, sin emplear modelos que requieran ser entrenados o calibrados. En [9] trabajan con un corpus compuesto por tweets, los cuáles analizan a nivel sintáctico y forman gramas a partir de las dependencias sintácticas en cada dato del corpus.

En [10] utilizan varios tipos de modelos de aprendizaje profundo, siendo un modelo GRU a nivel de artículo el que logra mejores resultados para la tarea de AA. Además, trabajan también la verificación de autoría, que consiste en determinar si dos entradas pertenecen a la misma categoría, por medio de una red siamés (una red siamesa consiste en dos subredes con la misma arquitectura, parámetros y pesos).

3. Conjunto de datos

La presente sección presenta la descripción de los conjuntos de datos utilizados. En la sección 3.1 se detalla el contenido y creación del corpus de autores literarios y la sección 3.2 describe el corpus de tweets en español.

3.1. Corpus TLE

El corpus de Textos Literarios en Español, denominado corpus TLE, ha sido construido para la realización de este trabajo debido a que, al menos hasta el momento de comenzar el presente proyecto, no se encontró un corpus adecuado para esta tarea. Este corpus TLE se formó recolectando y transcribiendo diversas obras de nueve autores literarios hispanohablantes de finales del siglo XX y principios del XXI, entre las que se incluyen novelas, cuentos y ensayos.

El corpus puede ser proporcionado a la comunidad científica solicitándolo directamente a los autores del artículo. Aproximadamente, el corpus se compone de 1,300,000 palabras, sin contar signos de puntuación. Sin embargo, el análisis que se hace sobre estos datos se enfoca en el manejo de párrafos como subestructura de los textos, entonces se dividen las obras en párrafos (cada diálogo se consideran un párrafo) y se les asigna la correspondiente etiqueta de acuerdo con el autor de cada párrafo.

El corpus TLE se divide en tokens como parte del preprocesamiento y es de interés observar la distribución de la longitud de los párrafos para poder definir un tamaño máximo del párrafo, medido en la cantidad de tokens, ya que es necesaria la uniformidad de esta característica para los modelos de aprendizaje profundo utilizados. Para esto se realiza un análisis sencillo de la frecuencia de longitud Fig. 1, tras lo cual se realizan pruebas con un modelo preliminar utilizando diferentes rangos de longitud de párrafo, con el fin de obtener una idea de la utilidad aparente de los rangos de párrafos.

Tras esto, se decide trabajar con un subconjunto del corpus que se componga de aquellos párrafos con longitud entre 50 y 500 tokens, el resto son despreciados por considerarse o carentes de información (en el caso de aquellos con menos de 50 tokens) o puntos atípicos (aquellos con longitud mayor a 500). En la Tabla 1 se describe el número de párrafos por cada autor en esta nueva versión del corpus TLE.

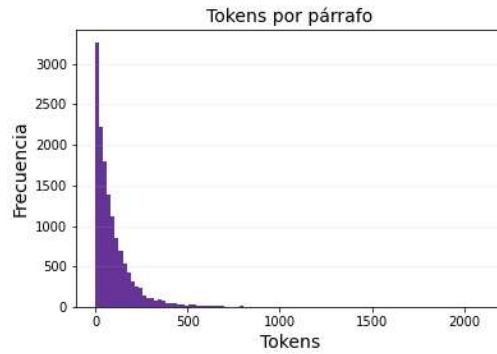


Fig. 1. Frecuencia del tamaño de tokens en cada párrafo del corpus TLE.

Tabla 1. Cantidad de párrafos por autor en el corpus TLE.

Autor	Número de párrafos	Autor	Número de párrafos
Autor 1	862	Autor 6	1240
Autor 2	1091	Autor 7	621
Autor 3	711	Autor 8	1133
Autor 4	765	Autor 9	740
Autor 5	712		
Total de párrafos: 7875			

Como se observa en la Tabla 1, el número de párrafos por autor no es equitativo para cada uno, por lo que basarse meramente en la precisión general del eventual modelo podría llevar a mal interpretación. Para mitigar esta problemática, se pueden utilizar otras métricas de evaluación del modelo, como f1-score que toma en cuenta tanto la cantidad de errores como el tipo de estos que comete el modelo en sus predicciones.

3.2. PAN

PAN es una serie anual de eventos científicos sobre textos forenses y estilometría. En este trabajo se utiliza únicamente la porción del corpus encontrado en [3] correspondiente al idioma español con las etiquetas de género del autor. Se realiza un análisis sobre el corpus con el mismo fin de fijar una longitud máxima de tokens. Así, se observa que casi todos los tweets tienen una longitud de 50 o menos tokens, y hay menos de 100 tweets con longitud mayor a 50, por lo cual, se decide excluirlas.

En este caso, no se establece de antemano una longitud mínima. Como se mencionó, al trabajar con este conjunto de datos se utilizan las etiquetas de género, por lo que el problema de AA en este caso se reduce a clasificación binaria (0 para femenino y 1 para masculino), además, el corpus ya está balanceado, es decir, la distribución de las etiquetas (hombre y mujer) es aproximadamente 50:50, por lo que de inicio, sabremos

que un modelo con precisión mayor a 50% es en sí mejor que predecir por mera probabilidad.

4. Metodología propuesta

Para cada corpus descrito en la sección anterior, se realizan experimentos con la metodología mostrada en la Fig. 2 y descrita a continuación.

1. Obtención del corpus. Esta etapa consiste en la recolección de los datos para formar los corpus TLE y PAN, descritos en la sección 3.1 y 3.2 respectivamente, y la división en 3 subconjuntos correspondientes al conjunto de entrenamiento (80%), validación (10%) y prueba (10%). Los conjuntos de entrenamiento y validación se utilizan durante la etapa de entrenamiento de los modelos de aprendizaje profundo, mientras que el conjunto restante se utiliza para probar el modelo en una situación que se considera un caso de uso real y es este resultado el que se considera para evaluar la utilidad del modelo. Además, se realiza un breve análisis de los datos recolectados a fin de filtrar aquellos de los que se pueda prescindir.
2. Tokenización. Consiste en segmentar el corpus en tokens (en este proyecto se utiliza la librería spaCy) con el fin de facilitar el trabajo en la etapa siguiente.
3. Análisis de los datos. El objetivo de esta fase es obtener conocimiento sobre la composición de los datos en los corpora, por ejemplo la riqueza del vocabulario, categoría gramatical de los tokens, etc. La profundidad del análisis se limita a la información gramatical que asocia spaCy a los tokens.
4. Preprocesamiento. Con base al análisis en la etapa previa, se ejecuta un conjunto de técnicas comunes de preprocesamiento (lematización, conversión a mayúsculas/minúsculas, remoción de stop-words, etc) sobre los conjuntos de entrenamiento y validación, se extrae el vocabulario asociado a estos por medio de un diccionario y se mapea cada token a su valor índice en el diccionario (las etiquetas de cada corpus igualmente se convierten a alguna representación numérica). Finalmente se trunca o rellena el párrafo para ajustar su longitud a la máxima longitud fijada en la primera etapa.
5. Construcción del modelo. En esta etapa se deciden los hiperparámetros de los modelos empleados (RNC, RNR y transformer), posteriormente se construye el modelo y finalmente se le entrena utilizando los conjuntos de entrenamiento y validación previamente preprocesados. Para ambos corpora, se emplea una capa Embedding entrenable propia del problema, es decir, no se utiliza un modelo de embedding pre-entrenado. Esto con la finalidad de que los valores del vector estén además asociados a la resolución de la tarea.
6. Evaluación del modelo. El modelo con mejores resultados sobre el conjunto de validación del paso anterior se testea con el conjunto de pruebas para tener una idea del desempeño del modelo bajo un caso de uso real, utilizando una métrica que mejor se adapte a cada corpus utilizado. Los resultados obtenidos se comparan con modelos previos para tratar de obtener algún indicio de

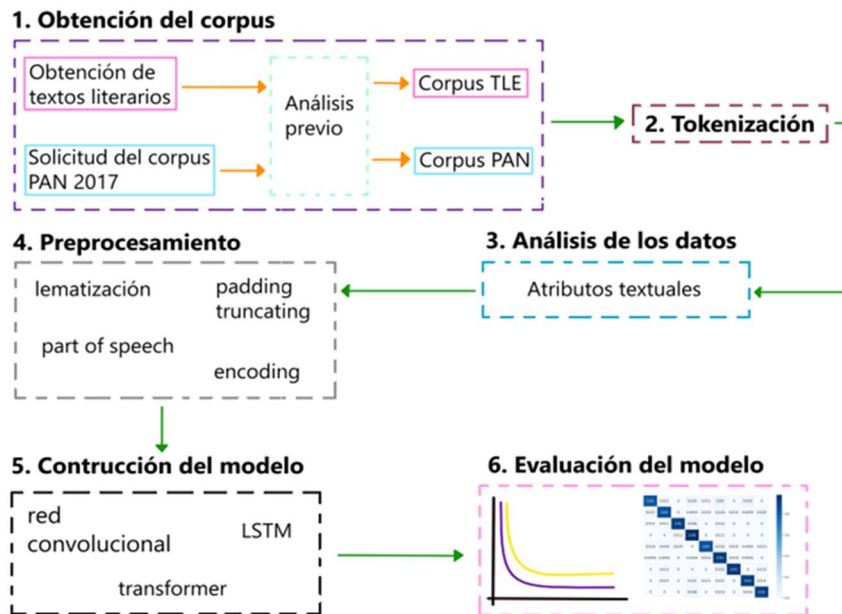


Fig. 2. Metodología propuesta para la atribución de autoría en textos en español.

cuáles características textuales son de mayor o menor utilidad para resolver la tarea de AA para el corpus correspondiente.

Esta metodología permite comparar diversos preprocesamientos y modelos; además, requiere una cantidad de recursos y tiempo grande debido a que básicamente se realiza un nuevo experimento por cada variación deseada. En este trabajo, esto no representa un inconveniente pues los tiempos de preprocesamiento y entrenamiento con los corpora utilizados son manejables, con recursos gratuitos -como Google Colab- son suficientes. Sin embargo, para proyectos de mayor tamaño este acercamiento tipo *prueba y error* no es viable.

5. Análisis y resultados experimentales

Esta sección se dedica a la exposición y discusión de los mejores resultados obtenidos para cada conjunto de datos. En las secciones 5.1 y 5.2 se presentan los resultados para los corpus TLE y PAN respectivamente, en la sección 5.3 se analizan los resultados en las dos secciones previas.

5.1. Corpus TLE

Se utiliza un modelo (TransformerTLE) ligeramente modificado del transformer obtenido de los ejemplos encontrados en la página de la API Keras [11]. La arquitectura

Tabla 2. Arquitectura del modelo transformer para el corpus TLE.

TransformerTLE		
Capa	Output Shape	# parámetros
Input	[(None, 500)]	0
TokenAndPositionEmbedding	(None, 500, 100)	5,993,700
TransformerBlock	(None, 500, 100)	84,416
GlobalAveragePooling1D	(None, 100)	0
Dropout	(None, 100)	0
Dense	(None, 64)	6464
Dropout	(None, 64)	0
Dense	(None, 9)	585

Parámetros entrenables: 6,085,165

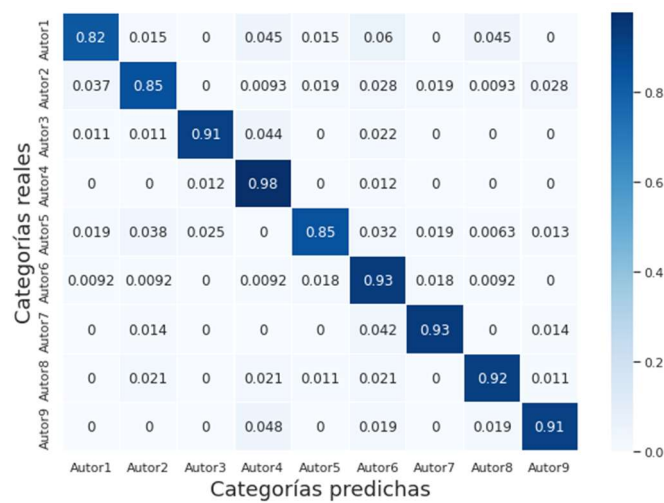


Fig. 3. Matriz de confusión del modelo transformer.

del modelo resultante se muestra en la Tabla 2. La capa *TokenAndPositionEmbedding* define internamente dos operaciones *embedding* (una para las palabras/tokens y otro para las posiciones de estas) resultando en vectores de 100 dimensiones.

La capa *TransformerBlock* define una capa *MultiHeadAttention* con 2 cabezas de atención y una capa *Dense* de 16 neuronas, con *relu* como función de activación, además de capas *LayerNormalization* y *Dropout* para optimizar las operaciones del modelo. A continuación, se toma promedio de las salidas del *Transformer* (*GlobalAveragePooling1D*) y se pasa por una red *feedforward*, que incluye una capa

Tabla 3. Arquitectura del modelo RNC para el corpus PAN.

RNC-PAN		
Capa	Output Shape	# parámetros
Embedding	(None, 50, 200)	82,128,800
SpatialDropout1D	(None, 50, 200)	0
Conv1D	(None, 48, 256)	153,856
GlobalMaxPooling1D	(None, 256)	0
Dense	(None, 1)	257
Parámetros entrenables: 82,282,913		

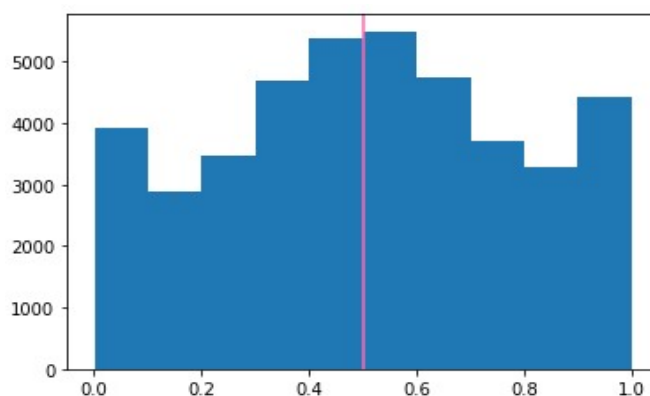


Fig. 4. Distribución de las predicciones del modelo RNC-PAN.

oculta Dense con 64 neuronas y relu como función de activación, para su clasificación (el output es producido por una capa Dense de 9 neuronas con activación softmax).

Las capas Dropout en todo el modelo utilizan un rate de 0.1. El modelo se entrena utilizando categorical-crossentropy como función de pérdida y el algoritmo adam como optimizador, además se emplea la técnica earllystop para detener el entrenamiento en cuanto no haya mejora (delta igual a 0.001) en el valor de la función de pérdida sobre el conjunto de validación y evitar así un posible caso de sobreentrenamiento (overfitting).

Las pruebas con el conjunto de pruebas arrojaron un resultado de 89.59% de precisión en las predicciones, la matriz de confusión asociada se muestra en la Fig. 3. Con el propósito de tener en consideración el desbalance de la cantidad de datos por autor (Tabla 1), se emplea la métrica f1-score, la cual da un resultado de 0.896.

5.2. Corpus PAN

Se construye un modelo sencillo de red convolucional (RNC-PAN) que se muestra en la Tabla 3. En este modelo la capa *Embedding* produce vectores de tamaño 200, a

continuación se utiliza una capa *SpatialDropout1D* con *rate* 0.1 y una capa *Conv1D* con 256 *kernels* con tamaño de ventana igual a 3 (*stride* por default igual a 1) y *relu* como función de activación. Finalmente se reduce la dimensión de la salida de esta capa por medio de *GlobalMaxPooling1D* y se clasifica utilizando una neurona binaria con función de activación *sigmoid*.

El modelo se entrena con la función de pérdida *binary_crossentropy*, optimizador *adam* y utilizando nuevamente *earlystop*. En el conjunto de pruebas, el modelo de la Tabla 3 logró una precisión del 73.10%. En la Fig. 4 se muestra la distribución de las predicciones del modelo (se muestra la probabilidad de pertenencia a cada clase, 0 o 1). Como métrica adicional a la precisión se mide el área bajo la curva ROC (o AUC-ROC por sus siglas en inglés), el cual resulta en 0.8187.

5.3. Análisis

Con base en los resultados mostrados en la Fig. 3, el modelo TransformerTLE logra un buen desempeño para clasificar textos de los nueve autores tratados, aunque para algunos tiene mayor conflicto para identificarlos. Esto podría deberse a mayor presencia de vocabulario no antes visto por la red en las porciones de esos autores en el conjunto de pruebas, a diferencia de sus contrapartes con mucho mayores resultados de precisión, o quizá el enfoque aplicado durante el análisis no es tan conveniente para ciertos autores como lo es para el resto.

Respecto al modelo RNC-PAN, sin tener del todo un terrible desempeño, el modelo no es capaz de discernir claramente, entre las dos categorías posibles (el mejor puntaje alcanzado durante la competencia PAN del 2017 para la misma porción del corpus fue de 83.21% de precisión). Aun así, el puntaje bajo la curva ROC alcanzado por el modelo es de 0.8187, indicando que el desempeño es medianamente bueno. Esto puede deberse, entre otros motivos, a que el análisis realizado no es el indicado para el tipo de contenido del corpus o a que el modelo propuesto, a pesar de ser el mejor de los que se probaron, está lejos de ser una arquitectura adecuada para el problema.

6. Conclusiones y perspectivas

El modelo TransformerTLE alcanza resultados satisfactorios para resolver la tarea de atribución de autoría para los 9 autores tratados en este trabajo. Sin embargo, el margen de mejora es aún considerable considerando que, como se muestra en la Fig. 3, hay autores para los que el modelo tiene más conflicto de identificar. Más aún, un análisis lingüístico más especializado sobre el corpus TLE podría llevar a mejores resultados. Sumado a esto, como se mencionó anteriormente, se utilizó un modelo basado en [9], por lo que una arquitectura más especializada o mejor diseñada para el corpus TLE resulte más conveniente.

El modelo RNC-PAN por otro lado, aunque no se acerca a los mejores resultados en la competencia, no obtuvo tampoco un pésimo desempeño y cumple medianamente con el objetivo de atribución de autoría (perfilado de autor específicamente). Es claro que puede mejorarse, al menos hasta el punto del estado del arte en la competencia del 2017,

quizá por medio de un análisis más exhaustivo de los datos o un enfoque distinto de análisis (por ejemplo, debido a que se lematizaron las palabras en este trabajo se perdieron los detalles de conjugación). De igual forma, queda abierto diseñar una arquitectura o modelo diferente.

Como trabajo futuro se puede resolver el problema con una perspectiva lingüística más profunda en el análisis de los datos a fin de obtener características más concretas que puedan llevar a mejores resultados, lo que a su vez podría llevar a reducir la complejidad de los modelos (es decir, la cantidad de parámetros entrenables).

Además, con una selección diferente de las características lingüísticas puede darse el caso que una arquitectura o un modelo diferente a las empleados en este trabajo sea más eficaz, lo que deja abierto el trabajo a desarrollar o adaptar un nuevo modelo de clasificación; aunque no necesariamente un modelo de aprendizaje profundo, ya que si bien han demostrado que pueden sobrepasar el desempeño de un modelo de aprendizaje automático clásico, no es una verdad aplicable a todo problema, tal como se observa en [1] y [2], por tanto, puede incluso darse el caso que un modelo clásico obtenga un mejor desempeño para la tarea tratada en este trabajo.

Referencias

1. Wang, H., Zhou L.: News Authorship Identification with Deep Learning (2016)
2. Macke, S., Hirshman, J.: Deep Sentence-Level Authorship Attribution (2015)
3. Rangel, F., Rosso, P., Potthast, M., Stein, B.: PAN17 Author Profiling [Data set]. In: CLEF 2017 Labs and Workshops, Notebook Papers. Conference title: PAN at Conference and Labs of the Evaluation Forum (2017).
4. López-Velasco, F.: Verificación de autoría en textos mediante redes neuronales convolucionales y recurrentes. Tesis de maestría, Universidad Nacional Autónoma de México, México (2018)
5. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: PAN13 Author Profiling [Data set]. In: CLEF 2013 Labs and Workshops, Notebook Papers. Conference title: PAN at Conference and Labs of the Evaluation Forum (2013)
6. Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., Daelemans, W.: PAN14 Author Profiling [Data set]. In: CLEF 2014 Labs and Workshops, Notebook Papers. Conference title: PAN at Conference and Labs of the Evaluation Forum (2014)
7. Martín-del-Campo-Rodríguez, C.: Atribución de autoría con aprendizaje automático. Tesis de maestría, Instituto Politécnico Nacional, México (2019)
8. Castro, D., Adame, Y., Pelaez, M., Muñoz, R.: Verificación de autoría, clasificación por vecindad. *Computación y Sistemas*, vol 21, no. 2, pp. 181–201 (2017)
9. Castillo-Velásquez, F. A, Martínez-Godoy, J. L., Torres-Falcón, M. P., Zavala de Paz, J. P., Becerra-Chávez, A., Rizzo-Sierra, J. A.: Atribución de autoría de mensajes de Twitter a través del análisis sintáctico automático. *Research in Computing Science*, vol. 149, no. 11, pp. 91–101 (2020)
10. Qian, C., He, T., Zhang, R.: Deep Learning based Authorship Identification (2017)

Fernando Hernández-Ibarra, Belém Priego-Sánchez, David Pinto

11. Text classification with Transformer—Keras,
https://keras.io/examples/nlp/text_classification_with_transformer/ (2022)