Las relaciones de dependencias como una característica del estilo de escritura.

Dependency relationships as a feature of writing style.

Germán Ríos Toledo (1).
Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez.
german.rt@tuxtla.tecnm.mx.

Luis Alberto Ríos Coutiño (2). Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez, luis.rc@tuxtla.tecnm.mx.

Raúl Paredes Trinidad (3). Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez, raul.pt@tuxtla.tecnm.mx.

Jorge Williams Figueroa Corzo (4). Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez,
jorge.fc@tuxtla.tecnm.mx.

Elfer Isaías Clemente Camacho (5). Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez,
elfer.cc@tuxtla.tecnm.mx.

Francisco de Jesús Suárez Ruiz (6). Tecnológico Nacional de México/I.T. Tuxtla Gutiérrez.
francisco.sr@tuxtla.tecnm.mx.

Artículo recibido en mayo 24, 2023; aceptado en junio 05, 2023.

Resumen.

Para desarrollar un sistema de análisis automático de estilo de escritura confiable, es necesario considerar factores importantes como el número de textos disponibles, así como la cantidad de palabras u oraciones que integran dichos textos. Otro desafío importante que ha marcado la pauta en los estudios de Atribución de Autoría es determinar qué tipo de características resulta más adecuada para representar los textos para obtener tasas de atribución de autoría más elevadas. Tradicionalmente los textos se representan por medio de características agrupadas en tres categorías: léxicas, morfologías y sintácticas. Este artículo se enfoca en el análisis de las características sintácticas llamadas relaciones de dependencias para evaluar su efectividad en un sistema automático de Atribución de Autoría basado en aprendizaje automático. Las relaciones de dependencia se obtuvieron mediante al analizador sintáctico Stanfod Parser, el cual divide el texto en oraciones en forma automática y genera las relaciones de dependencia entre pares de palabras. Después del análisis sintáctico, se conoce la frecuencia con la que un autor hace uso de las relaciones de dependencia en la composición de sus textos. Los experimentos realizados con algoritmos de aprendizaje automático mostraron que las características sintácticas proporcionan resultados muy favorables para algunos autores en las pruebas de clasificación, y que además brindan información importante sobre patrones de uso no perceptibles a nivel léxico o morfológico.

Palabras claves: Aprendizaje automático, atribución de autoría, características, relaciones de dependencia.

Abstract.

In order to develop a reliable automatic writing style analysis system, it is necessary to consider important factors such as the number of texts available, as well as the number of words or sentences that make up those texts. Another important challenge that has set the standard in authorship attribution studies is to determine which type of features are best suited to represent texts in order to obtain higher authorship attribution rates. Traditionally, texts are

represented by features grouped into three categories: lexical, morphological and syntactic. This paper focuses on the analysis of syntactic features called dependency relations to evaluate their effectiveness in an automatic machine learning based Authorship Attribution system. The dependency relations were obtained using the Stanfod Parser, which automatically splits the text into sentences and generates the dependency relations between word pairs. After parsing, the frequency with which an author makes use of dependency relations in the composition of his texts is known. Experiments with machine learning algorithms showed that syntactic features provide very favorable results for some authors in classification tests, and that they also provide important information about usage patterns not perceptible at the lexical or morphological level.

Keywords: Authorship attribution, dependency relations, features, machine learning.

1. Introducción.

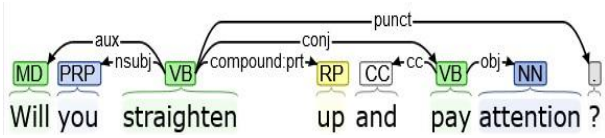
Uno de los puntos críticos de los sistemas de Atribución de Autoría basado en estilo es identificar características que permitan de identificar de forma precisa al autor de un texto anónimo. Con la llegada de los Analizadores Sintácticos como Stanford Parser, Spacy y Stanza es posible explorar con mucho detalle el papel que juega la información sintáctica de las oraciones en el estilo de escritura de un autor. Los Analizadores Sintácticos descomponen la estructura lineal de una oración y la muestran mediante una representación en forma de árbol invertido, en el que los nodos representan palabras que están conectadas mediante arcos. De acuerdo a Dearneffe et al (2008), las dependencias tipo Stanford se diseñaron para proporcionar una descripción simple de las relaciones gramaticales de una oración, de forma que personas sin conocimientos lingüísticos las comprendan fácilmente. Para explicar qué son las relaciones de dependencia, la Tabla 1 muestra las relaciones de dependencias de oraciones escritas en idioma inglés. El proceso de análisis sintáctico se realiza en cada oración del texto. El Analizador Sintáctico numera las palabras con al orden en que aparecen en una oración. Por ejemplo, en ”The lecture was really boring”, *The* es número 1, *lecture* es número 2 y así sucesivamente. Los signos de puntuación no son palabras, pero su posición en la oración también se numera. En las oraciones se usa una palabra *virtual ROOT* (en la posición 0) y una relación virtual *root*. Por ello, todas las oraciones comienzan con la relación de dependencia *root* (ROOT-0, X), donde X es la palabra que el analizador sintáctico identifica como la raíz de toda la oración.

Tabla 1. Análisis sintáctico de oraciones escritas en idioma inglés.

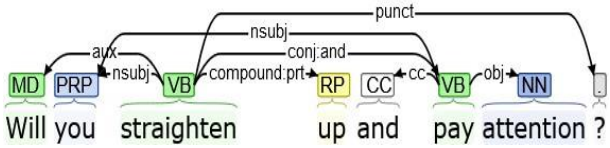
Oración	Relaciones de dependencia
The lecture was really boring.	root(ROOT-0, boring-5) det(lecture-2, The-1) nsubj(boring-5, lecture-2) cop(boring-5, was-3) advmod(boring-5, really-4) punct(boring-5, .-6)
The sonorous voice droned on.	root(ROOT-0, droned-4) det(voice-3, The-1) amod(voice-3, sonorous-2) nsubj(droned-4, voice-3) compound:prt(droned-4, on-5) punct(droned-4, .-6)
“Will you straighten up and pay attention?”	root(ROOT-0, straighten-4) punct(straighten-4, “-1) aux(straighten-4, Will-2) nsubj(straighten-4, you-3) nsubj(pay-7, you-3) compound:prt(straighten-4, up-5) cc(pay-7, and-6) conj:and(straighten-4, pay-7)

	obj(pay-7, attention-8) punct(straighten-4, ?-9) punct(straighten-4, ”-10)

Todas las relaciones de dependencias son binarias, Marneffe, et al, (2008); existe una palabra *gobernadora* y otra *subordinada*. Para la relación nsubj(boring-5, lecture-2), *boring* es palabra gobernadora y *lecture* es la palabra subordinada o dependiente. En esta forma de representación, la gobernadora siembre es la primera palabra. La información sintáctica también se representa en forma gráfica para una mejor compresión de las relaciones de dependencias. En un grafo dirigido los nodos representan palabras y los arcos a la relación de dependencia entre dos palabras, Sachan, (2020). La flecha señala a la palabra *dependiente*. Los analizadores sintácticos se actualizan periódicamente, por ello el conjunto de relaciones de dependencia y la forma de representarlas cambia de acuerdo a la versión. La figura 1 muestra las relaciones de dependencia de la oración: *Will you straighten up and pay attention?*



(a) Dependencias básicas.



(b) Dependencias extendidas

Figura 1. Relaciones de dependencias generadas con CoreNLP 4.4.0.

Además de las relaciones de dependencia, un analizador sintáctico proporciona más información importante sobre cada palabra de una oración. Por ejemplo, la categoría gramatical, así como el lema o entidades nombradas.

La Tabla 2 muestra clasificación de las relaciones de dependencia tipo Stanford propuesta en De Marneffe et al, (2014), los autores afirman que las relaciones se desarrollaron como una representación práctica de la sintaxis del inglés para aplicaciones de comprensión del lenguaje natural.

Tabla 2. Relaciones de dependencia tipo Standord.

Core dependents of clausal predicates		
Nominal dep	*Predicate dep*	
nsubj	csubj	
nsubjpass	csubjpass	
dobj	ccomp	xcomp
iobj		
Non-Core dependents of clausal predicates		
Nominal dep	*Predicate dep*	*Modifier word*
	advcl	afvmod
	nfincl	neg
nmod	ncmod	
Special clausal dependents		
Nominal dep	*Auxiliary*	*Others*

vocative	aux	marks
discuorse	auxpass	punct
expl	cop	
Coordination		
conj	cc	
Noun Dependent		
Nominal dep	*Predicate dep*	*Modifier word*
nummod	relcl	amod
appos	nfincl	det
nmod	ncmod	neg
Compounding and unanalyzed		
compound	mwe	goeswith
name	foreingn	
case-marking, prepositions,possesive		
case		
Loose joining relations		
list	parataxis	remnant
dislocated		reparandum
Other		
Sentence head	*Unspecified dependency*	
dislocated	dep	

2. Métodos.

2.1. Información del Corpus.

Para evaluar las relaciones de dependencias en Atribución de Autoría, se utilizó el Corpus del PAN 2012. Dicho Corpus consta de 14 autores de habla inglesa, cuyos nombres se identifican con caracteres de la A a la Z. El Corpus está dividido en un conjunto de entrenamiento y uno de prueba. En el conjunto de entrenamiento los autores cuentan con dos textos y en el conjunto de prueba con un solo texto. El problema de Atribución de Autoría consiste en determinar a quién de los 14 autores pertenece un texto cuya autoría "se desconoce". Este es un problema de clase "cerrada" dado que indudablemente uno de los N autores escribió el texto de prueba.

2.2. Preprocesamiento del Corpus.

Las Relaciones de dependencia se obtuvieron con el Analizador Sintáctico Starford Parser. Además del idioma inglés, dicho Analizador procesa textos escritos en español, francés, alemán, entre otros. Los árboles de dependencia representan las relaciones sintácticas entre las palabras que forman la oración, Posadas-Durán et al, (2017). La figura 2 muestra las relaciones de dependencia de una novela del conjunto de entrenamiento.

```
Sentence #10 (4 tokens):

His eyelids drooped.
Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, drooped-3)
nmod:poss(eyelids-2, His-1)
nsubj(drooped-3, eyelids-2)
punct(drooped-3, .-4)

Sentence #11 (8 tokens):

He began to slump in his seat.
Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, began-2)
nsubj(began-2, He-1)
nsubj:xsubj(slump-4, He-1)
mark(slump-4, to-3)
xcomp(began-2, slump-4)
case(seat-7, in-5)
nmod:poss(seat-7, his-6)
obl:in(slump-4, seat-7)
punct(began-2, .-8)

Sentence #12 (4 tokens):

“Dennis!”
Dependency Parse (enhanced plus plus dependencies):
root(ROOT-0, Dennis-2)
punct(Dennis-2, “-1)
punct(Dennis-2, !-3)
punct(Dennis-2, ”-4)
```

Figura 2: Relaciones de dependencias del PAN 2012.

2.3. Selección del tamaño de los textos.

Cada texto es distinto en tamaño en cuanto número palabras que los conforman. El analizador sintáctico divide el texto en oraciones de acuerdo a patrones aprendidos en múltiples entrenamientos. La Tabla 3 muestra el total de oraciones identificadas por Stanford Parser en los textos del corpus PAN 2012. Las celdas en negritas indican los textos con la menor cantidad de oraciones.

Tabla 3. Oraciones identificadas por Stanford Parser el corpus PAN 2012.

Conjunto de prueba		Conjunto de entrenamiento		
Texto anónimo	Oraciones	Autor	Oraciones Texto 1	Oraciones Texto 2
1	8,009	A	5,628	5,211
2	7,649	B	16,902	9,570
3	5,968	C	14,838	9,751
4	6,619	D	7,240	6,762
5	18,670	E	8,553	12,335
6	5,562	F	3,792	3,598
7	4,579	G	6,791	6,068
8	8,669	H	7,967	10,563
9	3,099	I	7,025	4,134
10	11,953	J	10,048	3,841
11	7,139	K	3,458	3,838
12	3,225	L	2,205	**2,723**
13	**1,637**	M	**1,781**	5,437
14	3,050	N	4,381	3,707

Debido a que los textos no tienen el mismo número de oraciones, es necesario normalizar la cantidad de bloques de texto utilizado para el entrenamiento de los algoritmos de aprendizaje. En los experimentos se determinó utilizar los primeros 10 bloques de cada novela. De esta forma todos autores disponen de la misma cantidad de bloques para el

entrenamiento. Situación que se conoce con el nombre de clases balanceadas. Además del número de bloques, también se definió la cantidad de oraciones a utilizar en cada uno de ellos. Los valores elegidos fueron 50, 100, 150 y 200 oraciones en cada bloque. La idea de modificar la cantidad de oraciones es analizar cómo influye la cantidad de oraciones en las pruebas de clasificación. Se espera que a mayor cantidad de información mejor será la exactitud del modelo generado por los algoritmos de aprendizaje. La Tabla 4 muestra la cantidad de texto en oraciones para el entrenamiento de acuerdo al tamaño del bloque.

Tabla 4. Total de oraciones para entrenamiento/prueba.

Número de bloques	Oraciones	Total de oraciones
10	50	500
	100	1,000
	150	1,500
	200	2,000

Cabe hacer notar que el autor M tiene 1,781 oraciones en el texto 1 de entrenamiento (ver Tabla 3), este autor solo contará con 8 bloques (en lugar de 10) con bloques de 200 líneas, este hecho no representa ninguna desventaja al realizar la atribución de autoría para dicho autor.

2.4. Matrices término-documento.

Una vez obtenidas las relaciones de dependencia, estas se contabilizan para obtener las frecuencias de uso en cada novela. Con las relaciones y frecuencias se generan matrices *termino-documento*. En estas matrices las filas representan los textos del autor y las columnas a las relaciones de dependencia. El analizador identificó 53 dependencias. La Tabla 5 muestra las 10 relaciones de dependencia más utilizadas en las novelas del conjunto de entrenamiento del corpus PAN 2012.

Tabla 5. Las 10 relaciones de dependencia de uso más frecuente del corpus PAN 2012.

Texto	Relaciones de dependencia en el conjunto de entrenamiento									
	punct	nsubj	case	det	root	advmod	obj	amod	dep	aux
01	16,358	11,929	9,979	9,835	8,010	7,530	5,300	5,206	3,045	3,320
02	16,010	11,085	9,116	8,711	7,652	7,324	5,226	5,980	2,816	3,712
03	15,371	9,836	7,983	7,072	5,969	6,982	4,717	4,894	3,167	3,377
04	17,812	7,726	5,149	5,456	6,619	4,546	3,501	3,113	3,100	1,539
05	46,193	28,346	17,319	15,650	18,671	15,323	12,093	7,962	8,267	7,435
06	16,036	9,366	7,616	7,681	5,562	5,716	4,791	3,429	3,737	2,781
07	12,122	8,495	7,456	6,871	4,579	4,633	3,889	3,743	2,245	2,493
08	20,679	13,972	8,106	8,821	8,670	7,125	5,883	3,544	2,624	4,277
09	8,197	5,610	4,495	3,999	3,099	2,951	2,603	2,290	1,472	1,627
10	32,120	19,194	16,157	14,329	11,953	9,711	8,917	9,402	5,696	4,677
11	17,786	10,366	7,365	7,596	7,140	5,965	4,988	3,997	3,003	2,712
12	11,516	7,291	9,989	8,329	3,227	5,296	3,891	4,551	2,515	2,693
13	6,056	3,723	7,525	6,807	1,637	3,364	1,811	4,014	1,426	1,463
14	9,639	6,295	4,939	4,601	3,051	3,698	3,406	1,991	1,832	2,110

En el conjunto de entrenamiento se generaron 4 matrices de 280x53 y en el conjunto de pruebas 4 matrices de 140x53. Cada matriz tiene información de frecuencias para los bloques de 50, 100, 150 y 200 oraciones respectivamente. La figura 3 muestra que a medida que aumenta la cantidad de texto, las frecuencias de uso también se incrementan.

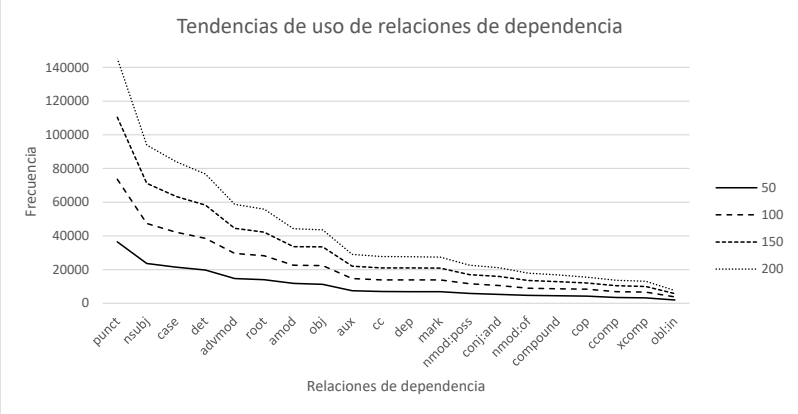


Figura 3. Tendencias de uso de las relaciones de dependencia.

2.5. Algoritmos de aprendizaje supervisado y algoritmos de reducción de dimensiones.

Para las pruebas de clasificación se utilizaron los algoritmos de aprendizaje automático supervisado Regresión Logística (LR, *Logistic Regresion*) y Vecinos Más Cercanos (K-NN, *k-Nearest Neighbors*). Con Regresión Logística se aplicaron los algoritmos de reducción de dimensiones Eliminación Recursiva de características con Validación Cruzada (RFECV, *Recursive Feature Elimination with Cross-Validation*) y SelectKBest. En la reducción de dimensiones, el algoritmo selecciona un subconjunto de variables más importantes de las 53 relaciones disponibles. El número de dimensiones se define de forma manual o se permite que el algoritmo determine el valor óptimo. Todos los algoritmos son implementaciones de scikit-learn y se reporta la métrica exactitud (*accuracy*) para validar la eficiencia de los clasificadores.

3. Resultados.

De acuerdo a Scikit-learn, RFECV es un estimador de aprendizaje supervisado con un método de ajuste que proporciona información sobre la importancia de las características. Su configuración fue RFECV (modelLR, step=1, cv=5) *step* es el número de dimensiones a eliminar en cada iteración, *cv* es la validación cruzada. El algoritmo determinó el valor óptimo de dimensiones. SelectKbest se configuró como SelectKBest (f_classif, k=25). Donde *f_classif* corresponde a ANOVA F-value y *k* al número de dimensiones. Además de *f_classif* también se evaluó *chi2* (CHI-cuadrado) pero con resultados no favorables. Con el clasificador K-NN, se utilizó el método GridSearchCV() para identificar los mejores parámetros en los bloques de 200 oraciones. Los parámetros evaluados se muestran en la figura 4. Despues de ese proceso los mejores parámetros identificados en la figura 5. De este modo, K-NN se configuró como *KNeighborsClassifier(n neighbors=12, p=1, weights='distance')*.

```
params = {
    'n_neighbors':  range(1, 15, 1),
    'p': [1,2],
    'weights': ['uniform', 'distance']
}

clf = GridSearchCV(
    estimator=KNeighborsClassifier(),
    param_grid=params,
    cv=5,
    n_jobs=5,
    verbose=1,
)
```

Figura 4. GridSearch para KNN.

```
Fitting 5 folds for each of 56 candidates, totalling 280 fits
{'n_neighbors': 12, 'p': 1, 'weights': 'distance'}
Traceback (most recent call last):
```

Figura 5. Mejores parámetros para KNN.

La tabla 6 muestra los porcentajes de exactitud en las pruebas de clasificación de los 14 autores. Cabe aclarar que aleatoriamente, la probabilidad de atribuir correctamente un texto al autor correcto es de 1/14 (aproximadamente 7.1%), una probabilidad muy baja debido a las 14 clases. Por ello es de suponer que el porcentaje de exactitud de los clasificadores difícilmente excederá el 50%. A continuación, analizaremos los resultados del algoritmo de Regresión Logística. En la primera prueba no se realizó la reducción de dimensiones, por lo que se utilizaron las 53 relaciones para generar el modelo. Se observa que, al incrementar el tamaño del bloque de texto, la exactitud alcanzó un máximo de 44%. En las dos pruebas siguientes con Regresión Logística, se aplicó la reducción de dimensiones con RFECV y SelectKBest. En RFECV se permitió que el algoritmo eligiera de forma automática las dimensiones más importantes, la exactitud más alta se obtuvo en bloques de 200 oraciones y 36 dimensiones con 47%. En el caso de SelectKBest el mejor resultado fue 48% de exactitud utilizando solo 25 dimensiones en el mismo tamaño de texto. La reducción de dimensiones contribuyó a mejorar, aunque con poco margen la exactitud de clasificador. Respecto al clasificador K-NN, las pruebas de clasificación se realizaron con 12 vecinos más cercanos, logrando 36% de exactitud en los bloques de 200 oraciones.

Tabla 6. Porcentajes de exactitud en las pruebas de clasificación.

Oraciones	**LR**		**LR-RFECV**		**LR-SelectKbest**			**K-NN**
	Dimensiones	**Precisión**	**RD**	**Precisión**	**RD**	**Precisión**	**Vecinos**	**Precisión**
50	53	19	34	41	20	41	12	23
100		25	50	43	20	37		25
150		20	39	51	20	45		22
200		**44**	36	**47**	25	**48**		**36**

Con base en los resultados previos se puede inferir que casi la mitad de los autores obtuvieron porcentajes de atribución favorables. Para obtener más detalles, se determinó obtener las matrices de confusión de las pruebas de atribución de autoría únicamente en bloques de 200 oraciones, ya que con dichos bloques se obtuvieron los mejores porcentajes de exactitud. La matriz de confusión muestra en el eje vertical al autor real del texto y en el eje horizontal el autor que el modelo predijo. Los números en la diagonal principal representan los casos donde el valor real coincidió con la predicción. La figura 6 muestra que los autores H, I, J y N tuvieron 5 de 10 atribuciones correctas, los 5 restantes fueron asignadas a otros autores. Los autores D y M tuvieron respectivamente 7 y 9 atribuciones correctas. De los 14 autores, 6 logran resultados aceptables.

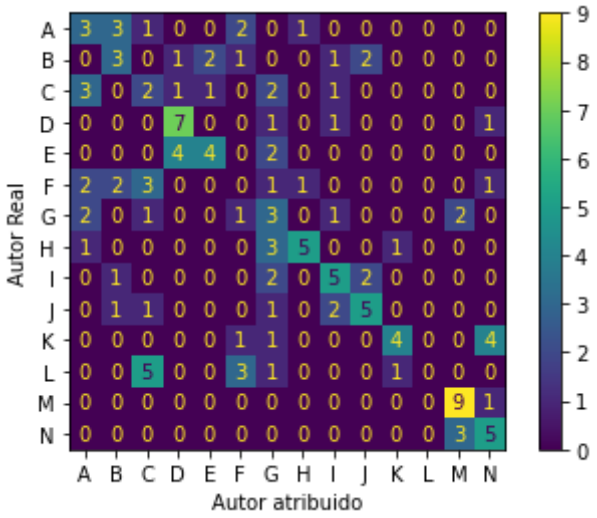


Figura 6. Matriz de confusión sin reducción de dimensiones.

La figura 7 muestra que, de forma individual, la reducción de dimensiones mejoró la exactitud de la clasificación. Los 10 textos de autores A y K se atribuyeron correctamente, 100% de exactitud a nivel individual. En ese mismo sentido, los autores C, E, H, J y M lograron de 7 atribuciones correctas de las 10 disponibles. Por otro lado, llama la atención que los autores B, F y L no lograron ninguna clasificación correcta. Todos sus textos se atribuyeron a autores distintos (7 textos de autor B se atribuyeron al autor J).

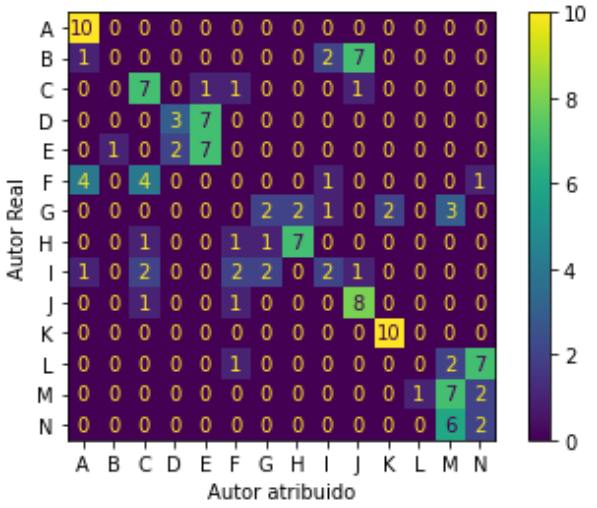


Figura 7. Matriz de confusión con RFECV.

En la figura 8 se muestra la clasificación con Regresión Logística y SelectKBest. Se observa la misma tendencia que en la figura 7, porcentajes de atribución muy buenos para autores como A, C, E, H, J, K y M. Nuevamente los autores B y F tuvieron 0 atribuciones correctas, pero en esta ocasión sus textos fueron atribuidos erróneamente a más de un autor de los restantes de la prueba.

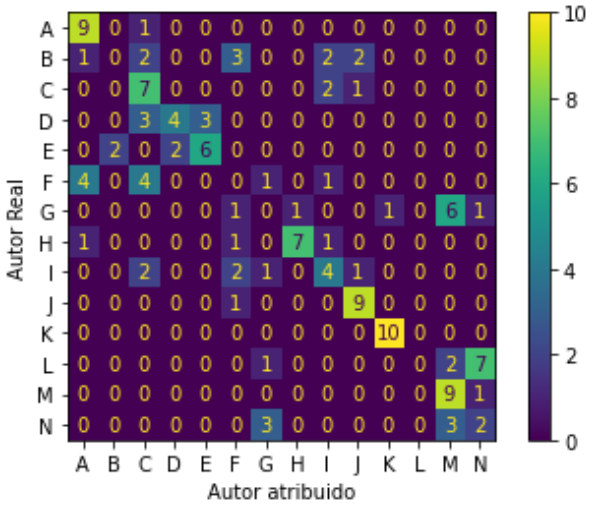


Figura 8. Matriz de confusión con SelectKBest.

Por último, la figura 9 muestra la clasificación con K-NN y 12 vecinos cercanos. Si bien el número de atribuciones correctas por autor no es mejor que las dos figuras previas, se mantiene la tendencia de la cantidad de autores con al menos 4 de 10 atribuciones.

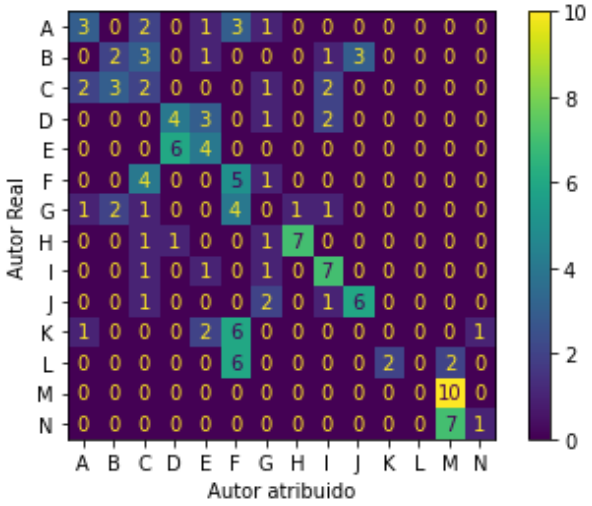


Figura 9. Matriz de confusión con KNN.

Conclusiones.

En este trabajo se investigó la utilidad de la información sintáctica en la tarea de Atribución de Autoría utilizando el Corpus PAN 2012, Jafariakinabad, F., et al, (2021). La Atribución de autoría consiste en identificar a quién de los 14 autores del Corpus pertenece un texto cuya autoría se desconoce. La tarea se abordó como un problema de clasificación con aprendizaje automático supervisado. Se seleccionaron los primeros 10 bloques de texto de cada autor, los bloques se formaron con 50, 100, 150 y 200 oraciones. El analizador sintáctico identificó 53 relaciones de dependencia. A partir de las relaciones se construyeron tablas de frecuencias de uso para cada uno de los autores. Las pruebas de

clasificación general mostraron porcentajes de exactitud inferiores a 50% tal y como era de esperarse. A mayor número de clases el porcentaje de clasificación correcta disminuye considerablemente.

Sin embargo, a nivel individual los resultados son distintos y favorables. Las relaciones de dependencia demostraron ser una característica útil en la atribución de autoría para algunos autores. A través de otras investigaciones, se sabe que la información sintáctica de las oraciones contiene patrones que un autor genera incluso inconscientemente a lo largo de sus textos. Cabe mencionar que, en este trabajo, las relaciones de dependencia identificadas se utilizaron de forma similar a un enfoque conocido Modelo de Bolsa de Palabras HaCohen-Kerner et al, (2020), Rustam, F., et al, (2021), Deepak, S., & Chitturi, B. (2020), Jafariakinabad, F., et al, (2022). El modelo de Bolsa de Palabras no se toma en cuenta la información contextual relacionada a cada palabra. Por ejemplo, qué palabras la preceden o suceden, no se considera el orden de las palabras, ni qué o cuales palabras coocurren con otras, En consecuencia, se desperdicia información valiosa sobre el estilo de escritura de un autor. Similarmente, en un modelo de "Bolsa de Relaciones de Dependencia" se ignora valiosa información sobre el contexto en que aparecen dichas relaciones ni qué palabras las conforman. A pesar de esos inconvenientes, en estos experimentos se demostró que las relaciones de dependencia mostraron por sí mismas (información puramente sintáctica sin ninguna información adicional), ser una opción viable para pruebas de atribución de autoría. Otros trabajos combinan distintos tipos de características como son las léxicas y morfológicas, Mehler, A. et al, (2018).

Referencias.

De Marneffe, M., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., & Manning, C. D. (2014). Universal Stanford dependencies: A cross-linguistic typology. In *Language Resources and Evaluation* (pp. 4585–4592). Springer Science+Business Media. https://nlp.stanford.edu/pubs/USD_LREC14_paper_camera_ready.pdf

De Marneffe, M., & Manning, C. D. (2008). Stanford typed dependencies manual. Tech. rep., Technical report, Stanford University.

Deepak, S., & Chitturi, B. (2020). Deep neural approach to Fake-News identification. *Procedia Computer Science*, *167*, 2236–2243. https://doi.org/10.1016/j.procs.2020.03.276

HaCohen-Kerner, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, *15*(5), e0232525. https://doi.org/10.1371/journal.pone.0232525

Jafariakinabad, F., & Hua, K. A. (2021). Unifying Lexical, Syntactic, and Structural Representations of Written Language for Authorship Attribution. *SN Computer Science*, *2*(6). https://doi.org/10.1007/s42979-021-00911-2.

Jafariakinabad, F., & Hua, K. A. (2022). A self-supervised representation learning of sentence structure for authorship attribution. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *16*(4), 1-16.

Mehler, A., Hemati, W., Uslu, T., & Lücking, A. (2018). A multidimensional model of syntactic dependency trees for authorship attribution. *Quantitative analysis of dependency structures*, 315-347.

Posadas-Durán, J. P., Sidorov, G., Gómez-Adorno, H., Batyrshin, I., Mirasol-Mélendez, E., Posadas-Durán, G., & Chanona-Hernández, L. (2017). Algorithm for extraction of subtrees of a sentence dependency parse tree. *Acta Polytechnica Hungarica*, *14*(3), 79-98.

Rustam, F., Khalid, M., Aslam, W., Rupapara, V., Mehmood, A., & Choi, G. S. (2021). A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. *Plos one*, *16*(2), e0245909.

Sachan, D. S., Zhang, Y., Qi, P., & Hamilton, W. (2020). Do syntax trees help pre-trained transformers extract information?. arXiv preprint arXiv:2008.09084.

Información de los autores.

Germán Ríos Toledo obtuvo el grado de Doctor en Ciencias de la Computación en 2019 por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) en Cuernavaca, Morelos, México. Actualmente, es profesor de tiempo completo en el Departamento de Computación del Tecnológico Nacional de México (campus Tuxtla Gutiérrez, Chiapas) en la Ingeniería en Sistemas Computacionales y en la Maestría en Ciencias en Ingeniería Mecatrónica. Su área de especialización es el Procesamiento del Lenguaje Natural, particularmente en el uso de información sintáctica como una característica para el análisis de estilo de escritura. Otras áreas de su interés incluyen el procesamiento y análisis de imágenes, audio y video por medio de Algoritmos de Aprendizaje Automático y Aprendizaje Profundo. Es Miembro del Sistema Nacional de Investigadores nivel Candidato en el periodo 2021-2024.

Luis Alberto Ríos Coutiño. Ingeniero en Sistemas Computacionales, graduado por Experiencia Profesional, desde hace 20 años ha trabajado para el sector público en Chiapas y privado para Empresas de Tecnología de Estados Unidos, Canadá, Colombia, Chile y México, esto en el ramo Financiero, Pago de servicios y de Logística, es profesor de asignatura en el Departamento de Ingeniería en Sistemas Computacionales del Tecnológico Nacional de México, campus Tuxtla Gutiérrez. Se ha especializado como Sysadmin DevOp configurando servicios en la nube, desarrollador de Aplicaciones Web como rol de Backend y Frontend, uso de base de datos, Ciencia de Datos e implementación de Inteligencia Artificial y Machine Learning en negocios.

Raúl Paredes Trinidad. Es egresado de la primera generación de la carrera de Ingeniería en Sistemas Computacionales del Tecnológico Nacional de México, Campus Tuxtla Gutiérrez. Estudió la Maestría en Tecnología Educativa en la Universidad Virtual del Instituto Tecnológico y de Estudios Superiores de Monterrey Cuenta con 23 años de experiencia como catedrático de medio tiempo del Tecnm y 28 años en nivel secundaria. Hace 14 años realizó una investigación sobre el desarrollo de los Clusters de TI en diferentes Estados de la República, tales como el Cluster de TI de Jalisco, el Cluster de TI de Querétaro y el Cluster de TI de Nuevo León, entre otros buscansdo impulsar el sector de TI en Chiapas. Es CEO de la empresa Ambar Rojo, desarrolladora de software a la medida, certificada en el año 2015 en Team Software Process (TSP). Desde hace 4 años y medio han desarrollado en tecnologías como: Angular, Asp net, Javascript, asp net webform, Sql Server, SAP commissions, entre otros. Ambar Rojo ha desarrollado proyectos de software a la medida como puntos de vetna, un sistema de trazabilidad para una empresa que administra flotillas, entro otros. Así mismo ha colaborado con empresa desarrolladoras de software de la CDMX y Baja California

Jorge William Figueroa Corzo obtuvo el grado de Doctor en Administración en 2016 por la Universidad Privada del Sur de México, en Tuxtla Gutiérrez, Chiapas, Actualmente es profesor de tiempo completo en el departamento de Sistemas y Computación del Tecnológico Nacional de México (campus Tuxtla Gutiérrez, Chiapas) en la ingeniería en Sistemas Computacionales. Su área de especialización es el comercio electrónico. Profesor con Perfil Deseable (PRODEP) desde 2020.

Elfer Isaías Clemente Camacho es Ingeniero en sistemas computacionales, por el Tecnológico de Tuxtla Gutiérrez, cuanta con Maestría en desarrollo de software por el Instituto de Estudios Superiores de Chiapas. Docente desde 2019 en la Carrera de Sistemas computacionales del Instituto Tecnológico de Tuxtla Gutiérrez. Docente en la Universidad Autónoma de Chiapas desde 2013.

Francisco de Jesús Suárez Ruiz obtuvo el grado de Maestro en Sistemas Computacionales con especialidad en Bases de Datos en 2010 por la Universidad Pablo Guardado Chávez, Tuxtla Gutiérrez, Chiapas, Actualmente es profesor de tiempo completo en el Departamento de Sistemas y Computación del Tecnológico Nacional de México (campus Tuxtla Gutiérrez, Chiapas) en la carrera de Ingeniería en Sistemas Computacionales, su área de especialización es la programación de propósito general, así como el desarrollo de aplicaciones móviles en lenguajes nativos IOS (Swift) y Android (Java). otras áreas de interés incluyen servicios Web, Redes y Bases de Datos. Miembro del Sistema Estatal de Investigadores, nivel II (2022-2024).