ISSN 1870-4069

Atribución de autoría de mensajes de Twitter a través del análisis sintáctico automático

Francisco Antonio Castillo Velásquez, José Luis Martínez Godoy, María del Consuelo Patricia Torres Falcón, Jonny Paul Zavala De Paz, Adela Becerra Chávez, José Amilcar Rizzo Sierra

Universidad Politécnica de Querétaro,
México

{francisco.castillo,jose.martinez,jonny.zavala,
consuelo.torres, adela.becerra}@upq.mx,
jose.rizzo@upq.edu.mx

Resumen. Dentro del Procesamiento del Lenguaje Natural (PLN) se ha considerado que la tarea de análisis sintáctico representa un alto costo computacional, por lo que su aplicación a varias tareas se ha visto limitada. Una de estas es la atribución de autoría de textos (AAT), que se encarga de responder a la cuestión de quién es el autor de un texto, dando algunos ejemplos previos de ese autor (conjunto de entrenamiento). A pesar del costo computacional, los trabajos de clasificación en este campo han dado buenos resultados para textos largos (por ejemplo, libros), pero el estudio de textos cortos ha quedado rezagado. En este trabajo de investigación se propone un modelo computacional basado en n-gramas sintácticos de dependencias para la AAT en Twitter. Como estos mensajes tienen una longitud corta (máximo 280 caracteres) el desempeño del parser sintáctico no es materia de preocupación. La metodología utilizada consistió en la compilación de tweets en español (corpus), su procesamiento en tareas de etiquetación de partes de oración (PosTags) para formar un baseline, la aplicación del análisis sintáctico de dependencias al corpus, la generación de gramas de PosTags y de dependencias sintácticas, la generación de archivos con datos de entrenamiento y clasificación y la aplicación de métodos supervisados de aprendizaje automático a estos archivos. Aunque la mayoría de los resultados no son alentadores, hay un conjunto que permiten ver la factibilidad de la AAT en mensajes de textos cortos.

Palabras clave: Atribución, Autoría, N-Gramas, Dependencias Sintácticas.

Attribution of Authorship of Twitter Messages through Automatic Syntactic Analysis

Abstract. Within the Natural Language Processing (NLP), it has been considered that the syntactic analysis task represents a high computational cost, so its application to various tasks has been limited. One of these is the attribution of authorship of texts (AAT), which is responsible for answering the question of who is the author of a text, giving some previous examples of that

author (training set). Despite the computational cost, classification work in this field has given good results for long texts (e.g., books), but the study of short texts has lagged. In this research work, a computational model based on syntactic n-grams of dependencies for the AAT in Twitter is proposed. As these messages are short in length (maximum 280 characters) the performance of the syntactic parser is not a matter of concern. The methodology used consisted of the compilation of tweets in Spanish corpus, their processing in tasks of tagging parts of speech (PosTags) to form a baseline, the application of the syntactic analysis of dependencies to the corpus, the generation of n-grams of PosTags and of syntactic dependencies, the generation of files with training and classification data, and the application of supervised methods of machine learning to these files. Although most of the results are not encouraging, there is a set that allows seeing the feasibility of AAT in short text messages.

Keywords: Attribution, Authorship, N-grams, Syntactic Dependencies.

1. Introducción

La atribución de autoría hace frente a una pregunta antigua y difícil: cómo asignar un texto de una autoría desconocida o disputada a un miembro de un conjunto de autores candidatos de quienes se tienen ejemplos de textos sin disputa [9]. A pesar de su aplicación a trabajos literarios, la rápida expansión de texto en línea en Internet (blogs, mensajes de correo electrónico, posts en redes sociales, etc.), revelan que las aplicaciones prácticas de la atribución de autoría son asociadas, por lo general, con trabajos forenses [1].

Los enfoques automatizados a este problema involucran el uso de métodos de aprendizaje o estadísticos [13]. Desde el punto de vista del aprendizaje automático, la atribución de autoría puede verse como una tarea de clasificación multi-clase y de etiquetación [11]. Existen dos etapas básicas: primero, los textos pueden representarse apropiadamente como vectores de valores numéricos y, luego, un algoritmo de clasificación puede usar estos vectores para estimar la probabilidad de asociación de un texto a una clase.

Desde entonces se han propuesto cientos de características estilométricas. Estas pueden distinguirse en las siguientes cinco categorías de acuerdo al análisis textual que requieren [13]: características léxicas (frecuencia de palabras funcionales, frecuencia de n-gramas de palabras, medidas de la riqueza del vocabulario, etc.), características de caracter (frecuencia de letras y n-gramas de carácter), características sintácticas (frecuencia de etiquetas POS, mediciones en la estructura de oraciones y frases, frecuencia de reglas de reescritura, etc.), características semánticas (mediciones de sinónimos, mediciones de dependencias semánticas, etc.) y características específicas de la aplicación (tamaño de fuente, color de fuente, frecuencias de palabras específicas, etc.). Hasta ahora, algunos estudios han demostrado que las mediciones más efectivas son las de características léxicas y de caracter.

El presente trabajo de investigación contribuye a la verificación de la factibilidad de aplicación del análisis sintáctico automático en el proceso de AAT para textos cortos, para nuestra ejemplificación, con mensajes de la red social Twitter.

2. Trabajo relacionado

Existen pocas referencias para trabajos sobre AAT en Twitter. En [19], los autores trabajaron con Ruby para obtener 393 características, como longitud de palabras, frecuencia de caracteres, palabras funcionales y longitud del texto, para procesarlas en clasificadores implementados en Matlab, con resultados de clasificación correcta no más allá del 40%.

Boutwell [22] realizó pruebas con un clasificador Naïve Bayes para n-gramas de caracter. La autora experimentó con 50 autores y dos conjuntos de entrenamiento (120 y 230). Complementó su trabajo con una serie de pruebas para estudiar el efecto de unir varios tweets en un solo documento.

Otro trabajo relacionado se encuentra en [21], donde los autores trabajaron con un conjunto de características que incluyen n-gramas de caracteres y de palabras, aunque en los experimentos solo se usó un tweet como documento para las pruebas (test). También trabajaron con los conceptos de *k-signature* y patrones flexibles que, al integrarlo con los n-gramas, lograron una mejora en sus resultados (hasta un 70% de clasificación correcta).

En [23] los autores propusieron un conjunto de marcadores estilísticos para la AAT para mensajes de Twitter.

Estos marcadores incluyeron emoticones, interjecciones, caracteres de puntuación, abreviaciones y otras características de bajo nivel. Solo trabajaron para tres autores con SVM, alcanzando una clasificación correcta del 63%.

3. Metodología usada

En este apartado se explica el modelo propuesto de trabajo, iniciando con la compilación del corpus, el proceso de obtención de los n-gramas y finalizando con la tarea de clasificación.

Los experimentos consistieron en la generación de n-gramas (mediante una herramienta libre de extracción); la compilación de los n-gramas hace referencia al almacenamiento dinámico de los n-gramas únicos (haciendo uso de estructuras matriciales y la implementación de un algoritmo para su manipulación); la refinación, que es la puesta a punto de los n-gramas únicos para que los caracteres no reconocidos por Weka sean detectados y cambiados; la generación de archivos Weka; la aplicación de estadísticas de frecuencias y la aplicación de procesos de clasificación con los modelos de SVM (SMO), Naïve Bayes y J48 para distintos tamaños de perfiles (profiles). Usamos un baseline tradicional con clasificación *NaïveBayes Multinomial Text* para el texto original como para etiquetas de *PoS*.

3.1. Compilación del corpus

La parte inicial del trabajo fue la compilación de un corpus de 600 tweets en español, intentando verificar la autenticidad de los mismos y seleccionando aquellos que no contuvieran caracteres de tipo emoticón. Este corpus está disponible para la comunidad investigadora.

Tabla 1. Rutas de dependencias para el grafo de la figura 2.

ao mod	cd v
ao cc conj	cd s
ao cc pass	cd cd spec
ao f	cd cd sp sn
cd conj	f
cd ci	

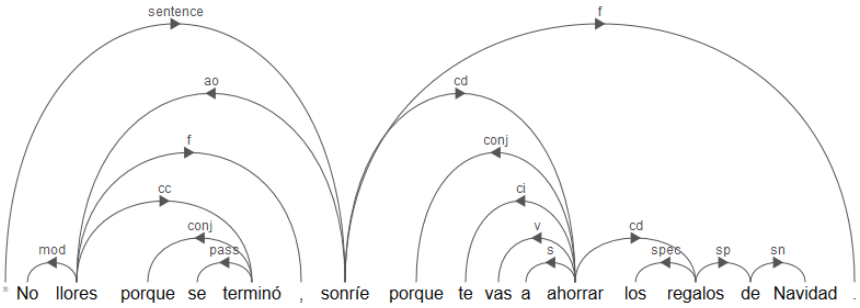

Fig. 1. Árbol de dependencias para una oración en español, generado por FreeLing.

```
1  No       no       RN      RN  pos=adverb|type=negative                                                  - - - 2  mod      - -
2  llores   llorar   VMSP2S0 VMS pos=verb|type=main|mood=subjunctive|tense=present|person=2|num=singular - - - 7  ao       - llorar.
3  porque   porque   CS      CS  pos=conjunction|type=subordinating                                       - - - 5  conj     - -
4  se       se       P00CN00 P0  pos=pronoun|gen=common|num=invariable                                   - - - 5  pass     - -
5  terminó  terminar VMIS3S0 VMI pos=verb|type=main|mood=indicative|tense=past|person=3|num=singular     - - - 2  cc       - termina
6  ,        ,        Fc      Fc  pos=punctuation|type=comma                                             - - - 2  f        - -
7  sonríe   sonreír  VMIP3S0 VMI pos=verb|type=main|mood=indicative|tense=present|person=3|num=singular - - - 0  sentence - sonreír
8  porque   porque   CS      CS  pos=conjunction|type=subordinating                                       - - - 12 conj     - -
9  te       te       PP2CS00 PP  pos=pronoun|type=personal|person=2|gen=common|num=singular              - - - 12 ci       - -
10 vas      ir       VMIP2S0 VMI pos=verb|type=main|mood=indicative|tense=present|person=2|num=singular - - - 12 v        - ir.00
11 a        a        SP      SP  pos=adposition|type=preposition                                         - - - 12 s        - -
12 ahorrar  ahorrar  VMN0000 VMN pos=verb|type=main|mood=infinitive                                      - - - 7  cd       - ahorrar
13 los      el       DA0MP0  DA  pos=determiner|type=article|gen=masculine|num=plural                    - - - 14 spec     - -
14 regalos  regalo   NCMP000 NC  pos=noun|type=common|gen=masculine|num=plural                          - - - 12 cd       - -
15 de       de       SP      SP  pos=adposition|type=preposition                                         - - - 14 sp       - -
16 Navidad  navidad  NP00000 NP  pos=noun|type=proper                                                    - - - 15 sn       - -
17 .        .        Fp      Fp  pos=punctuation|type=period                                            - - - 7  f        - -
```

Fig. 2. Análisis CONLL para la misma oración de la figura 2, generado por FreeLing.

A continuación, se muestran algunos ejemplos de tweets que forman parte del corpus. Se ha respetado la redacción original.

Algunas mentiras son de carne y hueso.

un clásico cuando haces una pregunta y como no saben que contestar te empiezan a insultar.

No llores porque se terminó, sonríe porque te vas a ahorrar los regalos de Navidad.

Cómo cambian las cosas!!

Cuando algo se me mete en la cabeza...imposible parar hasta que lo consigo

El dinero no puede comprar la felicidad, pero la pobreza no puede comprar nada.

3.2. Generación de n-gramas

Tomando como punto de referencia la segunda oración del ejemplo 1 (“*Cómo cambian las cosas!!*”) podemos generar los bigramas de caracteres *Có*, *óm*, *mo*, *o_*, *_c*, *ca*, *am*, *mb*, *bi*, *ia*, *an*, *n_*, *_l*, *la*, *as*, *s_*, *_c*, *co*, *os*, *sa*, *as*, *s!* y *!!*. También podemos generar los trigramas *Cóm*, *ómo*, *mo_*, *o_c*, *_ca*, *cam*, *amb*, *mbi*, *bia*, *ian*, *an_*, *n_l*, *_la*,

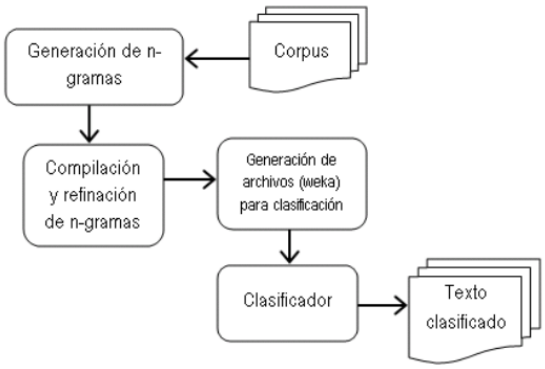

Fig. 3 Modelo de trabajo propuesto.

Tabla 2. Bigramas posibles para las rutas de dependencias de la tabla 1.

ao mod	ao f	cd cd
ao cc	cd conj	cd spec
cc conj	cd ci	cd cd
ao cc	cd v	cd sp
cc pass	cd s	sp sn

Tabla 3. Trigramas posibles para las rutas de dependencias de la tabla 1.

ao cc conj	cd cd sp
ao cc pass	cd sp sn
cd cd spec	

etc. La idea de trabajar con gramas es muy simple y tiene la ventaja adicional que puede aplicarse prácticamente para cualquier lenguaje.

Nuestro modelo hace un análisis estadístico de las apariciones de los gramas en cada una de las oraciones. Se pretende obtener un conjunto de características definitorias del estilo de un autor basado en este fundamento léxico de caracteres. Un análisis de este nivel (superficial) no necesita de un procesamiento profundo de las oraciones, como lo hace un análisis sintáctico (tanto de dependencias como de constituyentes).

El modelo del trabajo se resume en la figura 4, donde el paso inicial (generación del corpus) se dividió en dos conjuntos: los tweets originales y las dependencias generadas por el parser sintáctico FreeLing.

Para la misma frase de ejemplo, el parser de dependencias del Freeling regresa el árbol de dependencias de la Figura 2 y el análisis en formato CONLL (Figura 3):

La información CONLL es transferida a un módulo programado en Java para obtener todas las rutas de dependencias posibles. Para el ejemplo se obtendrían las rutas mostradas en la Tabla 1.

Estas rutas son procesadas con text2ngram [17] para obtener los gramas de dependencias. Por ejemplo, los bigramas posibles para el conjunto de la Tabla 1 se muestran en la Tabla 2 y para los trigramas en la Tabla 3 el único cuatrigrama posible es cd cd sp sn.

```
@relation autores
@attribute sn_sp_sn numeric
@attribute sp_sn_spec numeric
@attribute cc_sn_spec numeric
@attribute cc_sn_sp numeric
@attribute cc_sn_S numeric
... ...
@attribute cd_S_S numeric
@attribute cd_S_creg numeric
@attribute autores {marioruiz,mzavalag,rauljimenez,coffeejaay,egolepsia,lusagania}
@data
0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,2,
0,0,0,0,0,0,0,0,0,0,0,0,marioruiz
0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,marioruiz
... ...
1,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,lusagania
5,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,lusagania
```

Fig. 4. Ejemplo de archivo arff generado para su posterior clasificación.

En la generación de estos gramas, text2ngram realiza un análisis de frecuencias, lo que agrega un valor numérico al final de cada grama. Esta información (gramas y frecuencias) se almacenó en un archivo de texto para después cargarse y procesarse en una hoja de cálculo, obteniendo una ordenación de mayor a menor frecuencia. Una vez ordenados cada conjunto de gramas por autor, se realiza una recopilación de todos los gramas de todos los autores y se aplica de nueva cuenta una ordenación por frecuencia. De este conjunto final ordenado, obtenemos los gramas que pasarán a ser atributos en los archivos .arff, que en nuestro caso seleccionamos los primeros, 15, 30, 60 y 90 más frecuentes.

Estas primeras tareas las podemos fácilmente identificar en el modelo propuesto de trabajo de la Figura 4.

La implementación de la generación de archivos Weka es un proceso semi-automático a través de un programa en Java, el cual está disponible para la comunidad académica e investigadora [18], junto con el corpus compilado y los archivos Weka generados.

3.3. Proceso de clasificación

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Proporciona herramientas para el pre-procesamiento, clasificación, regresión, clustering y visualización de datos. Es un software de código abierto basado en los términos de GNU-GPL. En la Figura 5 se muestra un bosquejo de un archivo arff resultante, en el cual se identifican las tres secciones características de este tipo de archivos: nombre de la relación, atributos y datos.

Cada registro de la sección de datos (@data) representa un tweet y cada valor numérico representa las ocurrencias de un grama en particular en ese tweet. Por ejemplo, la primera terna de valores del último registro (5,1,0) significa que los gramas

Tabla 4. Baseline para 100 instancias por autor en texto original y partes de oración (PosTags), con clasificación por Naïve Bayes Multinomial Text (clasificación con texto completo original).

# Aut	Autores	Texto original	PoSTags
3	CoffeeJaay egolepsia LuSagania	0.53	0.52
3	Soymarioruiz Mzavalagc Raul_Jimenez9	**0.67**	0.63
6	CoffeeJaay egolepsia LuSagania Soymarioruiz Mzavalagc Raul_Jimenez9	0.47	0.44

sn_sp_sn, *sp_sn_spec* y *cc_sn_spec* aparecen cinco, una y cero veces en el tweet, respectivamente. El último valor, que en nuestro ejemplo no es numérico, representa al autor del tweet.

El proceso de entrenamiento y clasificación también se llevó acabo con el software Weka, que proporciona una diversidad de métodos. En particular, fueron usados los clasificadores NaiveBayes, Optimización Mínima Secuencial (SMO - Support Vector Machines) y árboles de decisión (J48), ya que estos han mostrado buenos resultados en otros trabajos de investigación, como en [12]. La configuración de la clasificación fue con una validación cruzada de 10 iteraciones y un porcentaje de división de 2/3.

4. Resultados experimentales

Los experimentos fueron desarrollados sobre los datos de un corpus para el problema de atribución de autoría. Con el corpus de 600 tweets se hicieron dos pruebas: una con 300 y otra con 600, o su equivalente, para 3 y 6 autores, ya que se recopilaron 100 tweets para cada autor. En la tabla 4 se muestra la información del baseline, con clasificación (basada en la medida de Cross Validation) del algoritmo Naïve Bayes Multinomial Text para el texto original de los tweets como para las partes de oración.

En los resultados que se mostrarán a continuación, usamos el término "*profile size*" para representar los primeros n-gramas/sn-gramas más frecuentes; por ejemplo, un tamaño del profile de 30 significa que se usaron solo los primeros 30 n-gramas más frecuentes. Probamos varios umbrales para el profile y seleccionamos 4 de ellos, como se muestra en todas las tablas de resultados.

Cuando alguna celda de la tabla contiene NA (*not available*, *no aplica*) significa que nuestros datos fueron insuficientes para obtener el número correspondiente de n-gramas. Sucede solo con los bigramas, ya que en general hay menos bigramas que trigramas, etc. En estos casos el número total de todos los bigramas es menor que el tamaño del profile.

Tabla 5. Resultados de la clasificación para el corpus de 3 autores, 60 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	**3**	**4**
15	NB	0.48	0.38	0.30
	SVM	0.53	0.52	0.35
	J48	0.40	0.45	0.28
30	NB	0.64	0.47	0.42
	SVM	**0.70**	0.50	0.40
	J48	0.53	0.55	0.30
60	NB	**0.69**	0.48	0.47
	SVM	0.64	0.38	0.42
	J48	0.50	0.38	0.32
90	NB	0.62	0.48	0.42
	SVM	0.60	0.38	0.40
	J48	0.52	0.38	0.33

Tabla 6. Resultados de la clasificación para el corpus de 3 autores, 150 instancias.

tamaño del profile	clasificador	tamaño del n-grama		
		2	**3**	**4**
15	NB	0.50	0.45	0.45
	SVM	0.54	0.50	0.41
	J48	0.43	0.53	0.43
30	NB	0.51	0.47	0.45
	SVM	**0.60**	0.48	0.42
	J48	0.44	0.43	0.45
60	NB	0.53	0.51	0.45
	SVM	**0.59**	0.47	0.48
	J48	0.45	0.45	0.45
90	NB	0.51	0.49	0.49
	SVM	0.55	0.47	0.47
	J48	0.46	0.49	0.45

La tarea de clasificación consiste en seleccionar características para construir el modelo de espacio de vectores, algoritmos supervisados de entrenamiento y clasificación – decidir a qué clase pertenece el texto –en nuestro modelo de espacio de vectores. En este trabajo presentamos resultados para tres clasificadores: SVM (SMO), Naïve Bayes y J48.

En los resultados mostrados en la tabla 5, es interesante notar que los que representan una mayor exactitud son dados por los clasificadores SVM y Naïve Bayes para bigramas, alcanzando un 70% el primero de ellos; de hecho, los mejores resultados

Tabla 7. Resultados de la clasificación para el corpus de 3 autores, 300 instancias.

Tamaño del profile	Clasificador	Tamaño del n-grama		
		2	**3**	**4**
15	NB	0.49	0.45	0.43
	SVM	0.52	0.43	0.38
	J48	0.49	0.40	0.39
30	NB	0.50	0.46	0.42
	SVM	**0.57**	0.45	0.41
	J48	0.54	0.43	0.38
60	NB	0.54	0.47	0.40
	SVM	**0.57**	0.45	0.43
	J48	0.51	0.40	0.38
90	NB	0.56	0.49	0.40
	SVM	**0.59**	0.45	0.44
	J48	0.51	0.41	0.38

Tabla 8. Resultados de la clasificación para el corpus de 6 autores, 600 instancias.

Tamaño del profile	Clasificador	Tamaño del n-grama		
		2	**3**	**4**
15	NB	0.32	0.28	0.25
	SVM	0.34	0.29	0.23
	J48	0.31	0.29	0.24
30	NB	0.36	0.31	0.27
	SVM	0.36	0.30	0.27
	J48	0.30	0.26	0.24
60	NB	0.36	0.31	0.26
	SVM	**0.37**	0.31	0.26
	J48	0.31	0.28	0.24
90	NB	**0.37**	0.31	0.26
	SVM	**0.37**	0.32	0.28
	J48	0.36	0.28	0.22

están para estos dos clasificadores en bigramas, salvo el caso de NB con un profile de 15.

En la tabla 6 se muestran los resultados de clasificación con 150 instancias, es decir, 50 tweets para cada uno de los 3 autores. Los mejores resultados los arroja el clasificador SVM para un tamaño de profile de 30 y 60 bigramas.

En la tabla 7 se muestran los resultados de clasificación para 3 autores, 100 tweets de cada uno de ellos, dando un total de 600 instancias. De nueva cuenta el clasificador

SVM obtiene los mejores resultados para cualquier tamaño de profile. Con las tablas 5-7, que representan los corpus de 3 autores, en general hay un mejor comportamiento con SVM, aunque las cifras no son tan alentadoras como se esperaban. La clasificación más alta está entre 57%-60%, para bigramas.

En la tabla 8 se muestran los resultados de clasificación para el corpus más grande compilado: 6 autores, 100 tweets para cada uno de ellos, dando un total de 600 instancias. Los resultados bajan drásticamente, no dando posibilidad de augurar mejores resultados con más autores y/o registros.

En este caso, las cifras del baseline (Tabla 4) resultan mejores, ya que alcanzaron un 47%, contra un 37%, que fue el mejor resultado para los gramas de dependencias sintácticas.

Se desarrolló una interfaz para la demostración del método, donde es posible configurar la elección de archivos a compilar, la frecuencia y tamaño de los n-gramas, entre otros parámetros.

Aunadas a estas aportaciones científicas, las de carácter técnico son la creación de una herramienta estadística para visualizar diversos resultados de los procesos involucrados y el corpus procesado para los experimentos (disponible para la comunidad científica [18]).

5 Conclusiones y trabajo futuro

Los resultados de los experimentos demuestran la factibilidad de usar modelos computacionales simples para la tarea de atribución de autoría de textos cortos, como correos electrónicos, tweets, mensajes de chats, etc., aunque solo para un conjunto reducido del corpus y n-gramas pequeños (la clasificación correcta más alta fue de 70% para solo 3 autores). La contribución más importante es la simplicidad y la generalidad del modelo (puede ser aplicados a cualquier lenguaje). Como trabajo futuro se pretende trabajar con la definición de nuevas características que, aunadas a los gramas de rutas de dependencias, mejoren substancialmente los resultados de clasificación; usar datasets estándares, como los que proporciona PAN-CLEF, para poder comparar nuestro método con otros que ya los han usado; y por último, probar otros métodos de clasificación, como Random Forest.

Referencias

1. Abbasi, A., Chen, H.: Sentiment analysis in multiple languages: feature selection for opinion classification in web forums. ACM Trans. Information Systems, 26(3), pp. 9−12 (2008)
2. Argamon, S., Levitan, S.: Measuring the usefulness of function words for authorship attribution. In: Proc. of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing (2005)
3. Agarwal, A.: Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-Gram. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL), pp. 24−32 (2009)
4. Baayen, H.: Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. Literary and Linguistic Computing, pp. 121−131 (1996)
5. Chaski, C.: Who wrote it? Steps towards a science of authorship identification. National Institute of Justice Journal, pp. 15−21 (1997)

6. De Marneffe, M.: Generating typed dependency parses from phrase structure parses. In: Proc. of LREC (2006)
7. Gamon, M.: Linguistic correlates of style: Authorship classification with deep linguistic analysis features. Proceedings of COLING 2004, pp. 611−617 (2004)
8. Hollingsworth, C.: Syntactic Stylometric: Using Sentence Structure for Authorship Attribution. Ms. Thesis, University of Georgia (2012)
9. Juola, P.: Questioned electronic documents: empirical studies in authorship attribution. In: Research Advances in Digital Forensics II Heidelberg: Springer. pp. 5−8 (2006)
10. López, A.: Atribución de autoría utilizando distintos tipos de características a través de una nueva representación. Tesis de maestría. INAOE, pp. 11 (2012)
11. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), pp. 1−47 (2002)
12. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic dependency-based N-grams as classification features. LNAI 7630, pp. 1–11 (2012)
13. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, pp. 538−566 (2009)
14. Stefanova, M.: El potencial discriminatorio de las secuencias de categorías gramaticales en la atribución forense de autoría de textos en español. Tesis doctoral, pp. 41−63 (2009)
15. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic N-grams as machine learning features for natural language processing. Expert Systems with Applications, 41(3), pp. 853–860 (2013)
16. Sidorov, G.: Construcción no lineal de n-gramas en la lingüística computacional: n-gramas sintácticos, filtrados y generalizados. 166 p. (2013)
17. N-Gram Extraction Tool: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/homepages.inf.ed.ac.uk/s0450736/ngram.html (2020)
18. Carpeta compartida de recursos del presente artículo: https://drive.google.com/ open?id=0 BwJ_YuKc8LgKWFgzc1JHYlhlVGs (2020)
19. Castro, A.: Author identification on twitter. In: Hardesty, Third IEEE International Conference on Data Mining, pp. 705−708 (2003)
20. Green, R.M., Sheppard, J.W.: Comparing frequency- and style-based features for twitter author identification. In: C. Boonthum-Denecke & G. M. Youngblood (eds.), Flairs Conference AAAI Press (2013)
21. Schwartz, R., Tsur, O., Rappoport, A., Koppel, M.: Authorship attribution of micro-messages. In: EMNLP Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pp. 1880−1891 (2013)
22. Boutwell, S.: Authorship attribution of short messages using multimodal features. Master's thesis, Naval Postgraduate School (2011)
23. Silva, R., Laboreiro, G., TimGrant, L., Oliveira, E., Maia, B.: Twazn me!!! Automatic authorship analysis of micro-blogging messages. In: Proc. of the 16th International Conference on Natural Language Processing and Information Systems, NLDB'11, pp. 161–168, Berlin, Heidelberg (2011)