Un nuevo enfoque basado en perfiles con aprendizaje de representaciones

Dario G. Funez[1], Marcelo L. Errecalde[1],
Leticia C. Cagnina[1,2]

[1] LIDIC, Universidad Nacional de San Luis, San Luis, Argentina.
[2] Consejo Nacional de Investigaciones Científicas y Técnica (CONICET), Argentina.
{funezdario,merrecalde,lcagnina}@gmail.com

Resumen Los Enfoques Basados en Perfiles (EBP's) han mostrado muy buen comportamiento específicamente en la tarea de atribución de autoría. Este trabajo tiene como finalidad extender al EBP empleando aprendizaje de representaciones. Para ello, se utilizará la gran flexibilidad de los mecanismos de coincidencia (matching) que proveen los embeddings. La similitud entre perfiles, en este caso, ya no considerará únicamente aquellas palabras que coinciden "exactamente", sino aquellas que son lo "suficientemente similares", de acuerdo a un umbral predeterminado. Este trabajo comprende un estudio exhaustivo comparativo empleando las colecciones Enron y CIAPPA, donde quedará probada la viabilidad y efectividad de nuestra propuesta en relación a enfoques de EBP clásicos como SPI y KRD empleando escenarios con diferentes métodos de embeddings, tales como Word2Vec, Fastext y Glove.

Palabras claves: Enfoque Basados en Perfil, Aprendizaje de Representaciones, Atribución de Autoría, Perfilado de Autor, Embeddings

1. Introducción

Los Enfoques Basados en Perfiles (EBP's) han sido aplicados exitosamente para tratar problemas de atribución de autoría [Stamatatos, 2009]. En general, en este problema, se dispone de un conjunto de autores con sus respectivos documentos de autoría indiscutida y un texto de autoría desconocida es asignado a un único autor elegido de los candidatos.

Los EBP's construyen un perfil para cada autor, con información obtenida de los documentos redactados por el mismo [Escalante *et al.*, 2011]. Sus principales ventajas son la sencillez y la eficiencia en la categorización de textos y, en general, son muy competitivos respecto a los enfoques más tradicionales basados en instancias (EBI's). Es importante observar que, si bien en sus orígenes los perfiles se asociaban con los autores de un texto, los EBP's pueden ser utilizados en cualquier problema de categorización de textos donde, en lugar de autores, tenemos clases arbitrarias.

Un problema que presentan los EBP's es cuando el vocabulario del texto de prueba, no es un subconjunto del vocabulario del perfil del autor que se quiera

comparar. Es decir, existen palabras en el texto de prueba que no coinciden con la del perfil del autor. EBP en sus fórmulas de similitud/distancia emplea la cantidad de palabras que coinciden *exactamente*. En caso que se usen sinónimos o palabras similares, no se tiene en cuenta que se refieren al mismo concepto porque son palabras lexicográficamente diferentes. Este inconveniente es denominado como problema de la *no coincidencia* (o *mismatching*) en el área de recuperación de la información [Lizarralde *et al.*, 2017]. Este se presenta cuando se comparan unidades léxicas (palabras o términos) que son atómicas e indivisibles y son iguales cuando existe una coincidencia exacta entre sus componentes. Estas son disímiles en cualquier otro caso.

Algunos trabajos previos se han centrado en capturar relaciones semánticas entre palabras, a través de representaciones de palabras que contengan esa información [Mikolov *et al.*, 2013a,Bojanowski *et al.*, 2017,Devlin *et al.*, 2018]. En el área de aprendizaje de representaciones (*representation learning*) se define que a partir de datos sin ningún preprocesamiento especial, se los puede proyectar a un espacio de representaciones vectoriales con más semántica implícita. Este espacio de representaciones es conocido como *embeddings*, y se lo ha aplicado exitosamente en áreas muy diferentes del Procesamiento del Lenguaje Natural (PLN) como la traducción automática [Zou *et al.*, 2013], parsing sintáctico [Weiss *et al.*, 2015], clasificación de textos [Kim, 2014] y *question answering* [Bordes *et al.*, 2014], entre otras. El significado del *embedding* de una palabra (en inglés, *word embedding*) es una representación de una palabra por medio de un vector, cuyas componentes contienen relaciones sintácticas y semánticas [Almeida y Xexéo, 2019]. Por ejemplo, las palabras *rey* y *reina* en una representación clásica de palabras serían distintas, pero sus embeddings estarían cercanos en el espacio vectorial proyectado.

La propuesta de este trabajo surge a partir de la anomalía de los EBP's, en cuanto al cálculo de la similitud entre los perfiles. Los embeddings de palabras contienen relaciones implícitas entre los mismos que no son actualmente consideradas en los perfiles de palabras. Teniendo en cuenta esta falencia, se planteó la hipótesis de que con la incorporación de embeddings de palabras al EBP, mejoraríamos su desempeño en diversas tareas de análisis de autoría: atribución de autoría, perfilado de autor y categorización por tópicos. Para ello, se probó la hipótesis anterior en diferentes colecciones (dos de atribución de autoría, dos de perfilado de autor y una de categorización por tópicos) pero por cuestiones de espacio sólo se muestran en el estudio experimental dos de ellas. Se seleccionaron dos colecciones representativas de las tareas de atribución de autoría y perfilado de autor, que difieren en el idioma de los textos (inglés y español), poseen diferente cantidad de documentos (362 versus 196), distinto número y balance de las clases (80 versus 2, desbalanceada versus balanceada). Se utilizaron 3 enfoques de embeddings (estáticos) clásicos como Word2Vec, Fastext y Glove.

El resto de este trabajo está organizado como sigue: en la Sección 2 se describen brevemente las tareas que se abordarán en el estudio experimental; en la Sección 3 se describen los EBP's y en la Sección 4 se enumeran diferentes formas de aprendizaje de representaciones. Luego, en la Sección 5, se explica la propuesta

y en la Sección 6, se detalla la experimentación con el nuevo método en dos colecciones distintas. La Sección 7 finaliza con algunas conclusiones y trabajos futuros.

2. Tareas de Categorización de textos

En esta sección se introducirán brevemente las dos tareas de Análisis de Autoría evaluadas en este trabajo: Atribución de Autoría (AA) y Perfilado de Autor (PA).

En atribución de autoría, se dispone de un conjunto de autores con sus respectivos documentos de autoría indiscutida y un texto de autoría desconocida es asignado a un único autor elegido de los candidatos [Stamatatos, 2009]. Los componentes de AA son: el conjunto de autores candidatos, un conjunto de documentos (conjunto de entrenamiento) escrito por algún autor candidato (todos los autores deben tener algún documento) y un conjunto de documentos (conjunto de prueba) de autoría desconocida que deben ser correspondidos con los autores candidatos.

En la tarea de perfilado de autor se pretende descubrir tanto como sea posible sobre un autor desconocido, analizando sólo el texto escrito por él [Rangel, 2013]. Así, características de los autores como por ejemplo género, edad, personalidad u orientación política, pueden ser descubiertas y, mediante el PA, clasificar textos de un autor desconocido según su perfil. Al igual que en AA, en PA se cuenta con un conjunto de textos de entrenamiento de varios autores que pertenecen a una clase particular que los caracteriza (el perfil) y se pretende clasificar documentos de prueba en base a los perfiles modelados.

3. Enfoques Basados en Perfiles

Los EBP's han sidos aplicados exitosamente para resolver problemas de AA [Stamatatos, 2009], siendo en diferentes momentos el estado del arte de esta tarea. El EBP consiste en construir un perfil para cada autor, con información obtenida de los documentos redactados por el mismo [Escalante *et al.*, 2011]. Para elegir el autor sobre los K autores candidatos, se utiliza la similitud o distancia entre el perfil del texto de prueba y el de cada perfil de los autores, y se elige el que resulte con mayor similitud o menor distancia. Si bien en los orígenes de los EBP's los perfiles se asociaban con los autores de un texto, también se pueden utilizar en cualquier problema de categorización de textos donde, en lugar de autores, tenemos clases arbitrarias y los perfiles se generan con los documentos de cada clase.

Para la generación de los perfiles de autor, se extraen de los documentos del autor un conjunto de L características. Para obtener el perfil de un autor, empleando por ejemplo la característica 3-gramas de caracteres, se recuperan todos los 3-gramas de todos los documentos del autor, y luego se los ordena en forma creciente por la cantidad de ocurrencias. Los n-gramas son subcadenas de n componentes consecutivos, estos pueden ser caracteres o palabras

[Cavnar y Trenkle, 1994]. El valor L es un parámetro del EBP y solamente se utilizan, para el caso del ejemplo, los $L's$ 3-gramas más frecuentes del perfil.

En el proceso de clasificación, lo primero que se debe realizar es la obtención del perfil del documento de prueba de autoría desconocida. Luego, mediante el uso de alguna medida de distancia o similitud, se debe comparar el perfil del documento de prueba con cada perfil de los K autores candidatos [Funez *et al.*, 2013]. La respuesta del clasificador es elegir el autor cuyo perfil es el más parecido al perfil del documento de prueba. Una componente principal en los EBP's es la de determinar la similitud/distancia entre los perfiles. La mayoría de los trabajos que han propuesto mejoras a los EBP's se centran en definir medidas de similitud más complejas y eficientes. A continuación se define la terminología que se empleará después en la definición de las medidas:

a) Se asume un escenario de K autores (clases) candidatos, con P_1 .. P_K perfiles correspondientes a los K autores (clases).
b) T_j denota el j-ésimo perfil de test.
c) La notación I_j^i significa el conjunto de términos que aparecen en la intersección de los perfiles P_i y T_j.
d) $f_X(n)$ denota la frecuencia de la característica n en el perfil X.
e) La medida denotada con S significa que es una medida de similitud y la expresada con D de distancia.

Las siguientes medidas son las más aplicadas en los EBP's:

a) Distancia Relativa de Keselj (Keselj's Relative Distance): Es un medida de distancia, también conocida como N-Gramas Comunes (CNG por sus siglas en inglés), en la cual la comparación de perfiles se realiza con una frecuencia normalizada de términos, como lo expresa la Ecuación 1 y se calcula respecto a los perfiles P_i y T_j [Keselj *et al.*, 2003]:

$$D_{krd}(P_i, T_j) = \sum_{n \in P_i \cup T_j} \left(\frac{2 \cdot (f_{P_i}(n) - f_{T_j}(n))}{f_{P_i}(n) + f_{T_j}(n)} \right)^2 \quad (1)$$

b) Similitud por Intersección de Perfiles Simplificada (Simplified Profile Intersection): Es una medida de similitud dada por la Ecuación 2 que calcula la cantidad de características que son comunes a ambos perfiles, sin aplicar ninguna normalización [Frantzeskou *et al.*, 2007]. En AA ha tenido mejor desempeño SPI con respecto a KRD.

$$S_{spi}(P_i, T_j) = \mid I_j^i \mid \quad (2)$$

4. Aprendizaje de representaciones

En los últimos años, en el PLN se ha investigado cómo capturar las relaciones semánticas entre las palabras y/o frases a través de representaciones más avanzadas [Almeida y Xexéo, 2019]. En el área del aprendizaje de representaciones se proyectan datos *crudos* a espacios de representación con mayor

semántica implícita. Este espacio proyectado se conoce como *embeddings* y son vectores densos de longitud fija que se obtienen usualmente mediante dos enfoques generales [Baroni *et al.*, 2014]: basados en *predicción* y basados en *conteo*. En los primeros, se aprende a predecir la probabilidad de ocurrencia del contexto de una palabra, usualmente mediante un enfoque de red neuronal (por ejemplo Word2Vec). Los embeddings (vectores) se derivan de los pesos de la red neuronal aprendidos en la resolución de esta tarea. Los modelos basados en conteo, en cambio, emplean información global recolectando estadísticas de la colección y el conteo de co-ocurrencias de palabras (por ejemplo Glove).
Otra diferencia importante de los embeddings es si éstos son *estáticos* (por ejemplo, Word2Vec, Fastext o Glove) o *contextuales* (por ejemplo, Bert). En los primeros se aprende un embedding fijo (único) por cada palabra/término en el vocabulario de embeddings. En los segundos, se aprenden embeddings contextuales dinámicos como la popular familia de representaciones BERT, en las que el vector para cada palabra es diferente en diferentes contextos. En las siguientes subsecciones se explican los modelos (de embeddings estáticos) Word2vec, Fastext y Glove que se utilizarán en la experimentación.

4.1. Word2Vec

Word2Vec entrena una red neuronal con textos y esta permite obtener a partir de una colección de documentos, los embeddings para cada palabra de la colección [Mikolov *et al.*, 2013b]. La intuición de Word2Vec es que entrenaremos a un clasificador en una tarea de predicción binaria: "¿Es probable que la palabra w aparezca cerca de una palabra objetivo?" En realidad, no nos importa esta tarea de predicción; en su lugar, tomaremos los pesos del clasificador aprendido como embeddings de palabras. El aspecto interesante, es que el texto bajo consideración actúa como datos de entrenamiento supervisados implícitamente para dicho clasificador (auto-supervisión), evitando la necesidad de cualquier tipo de señal de supervisión etiquetada a mano.
El modelo Word2vec puede usar uno de los siguientes tipos de arquitectura para el aprendizaje de los embeddings: CBOW o SG. En las dos arquitecturas se emplean redes neuronales con una única capa oculta, y en el peor de los casos tienen una complejidad de entrenamiento logarítmica lineal. La red neuronal utiliza el algoritmo *Retropropagación* (Backpropagation) para aprender los pesos de la capa oculta que darán origen a los embeddings de las palabras. Estos embeddings, pueden servir para derivar una representación de los documentos mediante alguna forma de agregación (usualmente el promedio) o como entrada para otras arquitecturas de redes neuronales más complejas (redes neuronales recurrentes, LSTM, etc).

4.2. Fastext

Este modelo de representación de palabras es una mejora de Word2Vec que toma en cuenta su *morfología* [Bojanowski *et al.*, 2017]. Se consideran como

unidades a las subpalabras y se representan las palabras por la suma de sus n-gramas de caracteres. En Fastext cada palabra w es representada como una bolsa de n-grams de caracteres, y se le agregan a cada palabra al principio el caracter $<$ y al final $>$, para distinguir los prefijos y sufijos de las demás secuencias de caracteres. La representación de una palabra es la suma de las representación vectorial de sus n-gramas.

4.3. Glove

Glove [Pennington *et al.*, 2014] es un algoritmo de aprendizaje no supervisado de embeddings de palabras cuyo entrenamiento se realiza sobre estadísticas globales de co-ocurrencia palabra a palabra recolectando estadísticas de la colección. Los embeddings obtenidos tienen subestructuras lineales importantes del espacio vectorial de palabras.
El modelo Glove se basa en que las diferencias vectoriales de los embeddings de las palabras, capturan lo mejor posible el significado de juntar ambas palabras. El entrenamiento del modelo Glove se lleva a cabo sobre un matriz global de co-ocurrencia palabra-palabra. La meta del entrenamiento de Glove es aprender los embeddings de las palabras, ajustando el producto punto con el logaritmo de la probabilidad de co-ocurrencia de las palabras. De esta manera, se produce la asociación del logaritmo de cocientes de probabilidades de co-ocurrencia, con la diferencia de vectores en el espacio vectorial de palabras.

5. Perfilado de autor con embeddings

Los embeddings de las palabras contienen información sobre relaciones semánticas entre las mismas que permiten identificar palabras "similares". En este trabajo se propone probar el efecto de modificar los perfiles para que, en lugar de contener las palabras (atómicas/indivisibles), se las sustituya por los embeddings (vectores) de estas palabras. Esto permite, mediante métodos como la similitud coseno entre vectores, reconocer aquellas palabras que se asemejan "lo suficiente" a una palabra específica. Para ello, se define un parámetro de umbral th, que indica el mínimo nivel de similitud que dos palabras deben tener para ser consideradas "semejantes" o "similares". Es claro, que cuando $th = 1$, sólo consideraremos como similares las palabras iguales.
Dado que los perfiles de palabras contienen las entradas de palabras con su frecuencia asociada, usaremos en los perfiles con embeddings una versión "ponderada" de la SPI que, al igual que la distancia KRD toma en cuenta la frecuencia de ocurrencia de las mismas. Por otra parte, ya no se considerará únicamente aquellas palabras que coinciden "exactamente" entre los perfiles, sino aquellas que son lo suficientemente similares, de acuerdo al umbral th.
A continuación se da el pseudo-código de esta implementación:

```
funcion sim_perf_embed(perfil_emb p1,p2)
acum = 0
Para cada e1 de p1
  Para cada e2 de p2
     sim = coseno(e1,e2)
     Si sim >= th
        acum += sim * frec(e2)
retornar acum
```

Como se puede observar, si no se tomara la frecuencia de las palabras y $th = 1$, sería la SPI ya descripta previamente. Esta nueva función de similitud entre perfiles, que denominamos *sim_perf_embed* recibe como parámetros dos perfiles de embeddings, el del documento de prueba (e1) y el de la clase (e2). Para cada par de embeddings de ambos perfiles, se computa su similitud coseno y, si este valor supera el umbral th, se incrementa la variable *acum*, de acuerdo a su similitud y su frecuencia en la clase. Finalmente el valor final *acum* es retornado por la función de similitud.

6. Evaluación experimental

En esta sección se describen las colecciones utilizadas en la experimentación, los resultados obtenidos y un análisis de los mismos.

6.1. Corpus Enron

Este corpus (*https://data.mendeley.com/datasets/n77w7mygwg/2*) es obtenido a partir del *Enron Email Dataset* [Halvani y Graner, 2018]. Es muy utilizado en investigaciones de la tarea de identificación de autoría. Está compuesto de 80 autores con emails que son textos planos formales escritos en inglés. Cada documento ha sido escrito por un único autor, y es una compilación de los más recientes emails producidos por cada uno de ellos. El conjunto de *entrenamiento* lo comprenden 282 documentos (80 autores, cada uno posee entre 2 y 4 documentos cada uno). El conjunto de *prueba* lo conforman 80 autores con un documento cada uno (80 documentos en total). En la tabla 1 se muestran los resultados de medida F para distintos L's con las métricas básicas SPI y KRD, y para el caso de perfiles con embeddings se usó la implementación de la función *sim_perf_embed* descripta previamente. De esta surgieron tres variantes: *sim_perf_W* que emplea Word2vec[1] para obtener los embeddings del corpus, *sim_perf_F* que usa Fastext[2] y *sim_perf_G*[3] que utiliza Glove. Para *sim_perf_W* se le eligieron los siguientes parámetros: tamaño de embedding de 12 y th = 0,99. *sim_perf_F* usó

[1] https://radimrehurek.com/gensim/models/word2vec.htmlmodule-gensim.models.word2vec
[2] https://radimrehurek.com/gensim/models/fasttext.htmlgensim.models.fasttext.FastText
[3] https://nlp.stanford.edu/projects/glove/

Enfoque perfiles			Enfoque perfiles con embeddings		
			Del Corpus		
L	*KRD*	*SPI*	*sim_perf_W*	*sim_perf_F*	*sim_perf_G*
50	0,1141	0,1154	0,1429	0,1419	0,1677
100	0,2752	0,2669	0,277	0,289	0,2702
200	0,3775	0,3642	0,3489	0,3654	0,3983
230	0,3749	0,3583	0,3748	0,3445	0,3935
250	0,3398	0,3435	0,3373	0,3648	0,3881
280	0,3895	0,402	0,4016	0,3862	0,4175
300	0,3714	0,3735	0,4283	0,3808	0,4659
330	0,4227	0,4235	0,4498	0,4252	0,4625
350	0,3873	0,3787	0,4148	0,4458	0,473
400	0,4095	0,4076	0,4519	**0,4713**	0,4402
500	0,4421	0,4402	0,4787	0,3876	0,4885
700	**0,5335**	**0,5335**	**0,5848**	0,4419	**0,5737**
800	0,4858	0,4691	0,5306	0,3983	0,5239
900	0,4658	0,4617	0,5047	0,3296	0,5181
1000	0,4817	0,4817	0,4863	0,2795	0,4697
1100	0,4146	0,4146	0,4713	0,2645	0,4727
1200	0,4084	0,4084	0,4879	0,2834	0,4997
1500	0,4265	0,4149	0,5004	0,2804	0,4997
1800	0,4265	0,4149	0,5004	0,2804	0,4997
2000	0,4265	0,4149	0,5004	0,2804	0,4997
2300	0,4265	0,4149	0,5004	0,2804	0,4997
2500	0,4265	0,4149	0,5004	0,2804	0,4997
3000	0,4265	0,4149	0,5004	0,2804	0,4997

Tabla 1. Medida F para el problema identificación de autoría Colección Enron.

un tamaño de embedding de 30 y th = 0,99. *sim_perf_G* usó un tamaño de embedding de 5, th = 0,9995, velocidad de aprendizaje = 0,05 y 30 épocas. Los parámetros fueron seleccionados luego de realizar una búsqueda exhaustiva de valores y se eligieron los que mejor comportamiento alcanzaron en la experimentación.

En la tabla 1 se encuentran resaltados los mejores resultados para los distintos enfoques probados y se puede observar que *sim_perf_W* alcanza la mejor medida F para $L = 700$ con $0,5848$ y los enfoques SPI y KRD obtuvieron su mejor medida F para $L = 700$ con $0,5335$. Así, *sim_perf_W* supera en un 9 % a SPI y kRD. Por otra parte, se puede observar que KRD no supera a SPI para ningún L. Respecto a los valores generales con los distintos L, *sim_perf_W* supera en un 86 % de los casos tanto a SPI como a KRD. También se puede apreciar que para L's mayores a 300 *sim_perf_W* supera a SPI en todos los casos. En particular, luego de $L = 1500$ no se tienen mejoras para los casos de las tres variantes. Por último, es claro que *sim_perf_F* alcanzó el peor resultado de las tres variantes con embeddings y *sim_perf_G* obtuvo una diferencia levemente inferior, considerando el mejor valor de la medida F, con respecto a *sim_perf_W*.

6.2. Corpus para la Identificación de la Alineación Política de Periodistas Argentinos (CIAPPA)

Este corpus está compuesto de 196 documentos pertenecientes a 10 periodistas, 5 de estos explícitamente apoyan las acciones del gobierno argentino que gobernó (entre los años 2012 y 2015), y los otros 5 son opositores al gobierno en ese periodo [Mercado *et al.*, 2019]. El corpus se divide en dos

grupos de documentos de acuerdo a la orientación política de los periodistas. De esta manera, 98 documentos pertenecen a la clase oficialista y 98 a la clase opositora, así el corpus es balanceado en sus dos clases. En la experimentación se planteó el problema de perfilado de autor para identificar la clase de un documento, y así determinar si el autor es oficialista u opositor al gobierno. Para el corpus CIAPPA no se encuentran disponibles los conjuntos de *entrenamiento* y *prueba*, estos se obtuvieron de forma aleatoria del corpus original. El conjunto de *entrenamiento* quedó compuesto por 84 documentos oficialistas y 84 opositores quedando balanceado en las dos clases, mientras que el conjunto de *prueba* lo comprenden 14 oficialistas y 14 opositores.

En la tabla 2 se muestran los resultados de medida F para los modelos basados en perfiles considerando los enfoques SPI y KRD y perfiles con embeddings. Para sim_perf_W y sim_perf_F se utilizaron los siguientes parámetros: el tamaño del embedding de 100 y th = 0,85. Para el caso de sim_perf_G se usó un tamaño de embedding de 5, th = 0,99, velocidad de aprendizaje: 0,05 y 30 épocas. Los mejores valores se muestran resaltados en la tabla 2 y se puede observar que sim_perf_W alcanza la mejor medida F para $L = 7000$ con $0,9282$ superando a SPI y KRD que obtuvieron $0,8916$.

Enfoque perfiles			Enfoque perfiles con embeddings		
			Del Corpus		
L	*KRD*	*SPI*	*sim_perf_W*	*sim_perf_F*	*sim_perf_G*
10	0,3333	0,3333	0,317	0,3333	0,3333
20	0,3333	0,3333	0,5618	0,3333	0,3333
50	0,4285	0,392	0,3437	0,4466	0,3858
70	0,747	0,747	0,5351	0,7857	0,8212
100	0,7083	0,747	0,6428	0,7417	0,6256
120	0,7846	0,7846	0,5692	0,7857	0,6679
150	0,747	0,7128	0,6066	0,7005	0,7005
180	0,747	0,747	0,63541	0,6111	0,6679
200	0,7846	0,78461	0,6781	0,6888	0,747
400	0,6781	0,6781	0,7496	0,6111	0,733
500	0,7083	0,7083	0,74968	0,7005	0,7812
600	0,8212	0,8212	0,7142	0,7005	0,8193
700	0,7857	0,7857	0,7857	0,7417	0,8212
750	0,8212	0,7857	0,8212	0,8212	0,8212
1000	0,8564	0,8564	0,8212	0,7812	0,8564
2000	0,747	0,747	0,8564	0,7857	0,8193
2300	0,8564	0,8564	0,8564	0,7496	0,8564
2500	0,8212	0,8212	0,8916	0,641	0,8564
2800	0,8564	0,8193	0,8564	0,7496	0,8564
3000	0,8564	0,8564	0,8564	0,7496	0,8564
3500	0,8564	0,8564	0,8564	0,6428	0,8564
4000	0,8193	0,8193	0,8541	0,641	0,8541
4500	0,8564	0,8564	0,8916	0,7128	0,8916
5000	0,8564	0,8564	0,8916	0,74176	0,8541
6000	0,7754	0,8155	0,8155	0,7417	0,8155
7000	**0,8916**	**0,8916**	**0,9282**	0,7812	**0,8916**
8000	0,8541	0,85416	0,9282	0,747	0,8541
9000	0,8193	0,8193	0,9282	0,747	0,8916
10000	0,8193	0,8193	0,89161	0,7417	0,8541

Tabla 2. Medida F corpus CIAPPA problema de perfilado de autor.

7. Conclusiones y Futuras Extensiones

Los EBP's han mostrado su eficiencia y buen comportamiento, particularmente en tareas de atribución de autoría. No obstante esto, se basan en criterios de comparación exacta entre palabras que podrían ser mejorados con los nuevos enfoques de aprendizaje de representaciones (embeddings).
Este trabajo, da una propuesta de cómo llevar a cabo esta tarea, proponiendo una función de similitud entre perfiles que toma en cuenta la similitud de sus embeddings constituyentes. Hasta donde sabemos, este es el primer trabajo en el área de los EBPs donde se propone una extensión con estas características.
Nuestra propuesta, obtiene mejores resultados que enfoques clásicos de EBP como SPI y KRD en las colecciones Enron y CIAPPA que difieren en el tipo de tarea, lenguaje y número y nivel de balance de sus clases. En ambos casos, se probaron embeddings estáticos clásicos como Word2Vec, Fastext y Glove obteniendo con los embeddings Word2Vec (variante Sim_perf_W) el mejor desempeño. Si bien por razones de espacio, sólo se reportan los resultados de estas dos colecciones, los resultados obtenidos con otras colecciones clásicas (como 20NewsGroup) han mostrado la viabilidad y efectividad de nuestra propuesta.
Como trabajo futuro se propone realizar un estudio experimental más exhaustivo con diferentes formas de ponderar la frecuencia de las palabras en los perfiles, combinación de perfiles con embeddings y perfiles con n-gramas de caracteres y adaptación de embeddings contextualizados (tipo BERT) al esquema de trabajo propuesto.

Referencias

[Almeida y Xexéo, 2019] Almeida, F. y Xexéo, G. (2019). Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.

[Baroni *et al.*, 2014] Baroni, M., Dinu, G., y Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. *ACL*, pp. 238–247.

[Bojanowski *et al.*, 2017] Bojanowski, P., Grave, E., Joulin, A., y Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

[Bordes *et al.*, 2014] Bordes, A., Chopra, S., y Weston, J. (2014). Question answering with subgraph embeddings. *CoRR*, abs/1406.3676.

[Cavnar y Trenkle, 1994] Cavnar, W. B. y Trenkle, J. M. (1994). N-gram-based text categorization. En *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pp. 161–175.

[Devlin *et al.*, 2018] Devlin, J., Chang, M.-W., Lee, K., y Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Escalante *et al.*, 2011] Escalante, H. J., y Gómez, M. M., y Solorio, T. (2011). A weighted profile intersection measure for profile-based authorship attribution. En *Proceedings of MICAI 2011*, volumen 7094, pp. 232–243.

[Frantzeskou *et al.*, 2007] Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C. E., y Howald, B. S. (2007). Identifying authorship by byte-level n-grams: The source code author profile (scap) method. *International Journal of Digital Evidence*, 6(1):1–18.

[Funez *et al.*, 2013] Funez, D. G., Cagnina, L., y Errecalde, M. L. (2013). Determinación de género y edad en blogs en español mediante enfoques basados en perfil. En *XVIII Congreso Argentino de Ciencias de la Computación*.

[Halvani y Graner, 2018] Halvani, O. y Graner, L. (2018). Rethinking the evaluation methodology of authorship verification methods. En *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 40–51. Springer.

[Keselj *et al.*, 2003] Keselj, V., Peng, F., Cercone, N., y Thomas, C. (2003). N-gram-based author profiles for authorship attribution. En *Proceedings of the Pacific Association for ComputationalLinguistics*, pp. 255–264.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification.

[Lizarralde *et al.*, 2017] Lizarralde, I., Rodriguez, J. M., Mateos, C., y Zunino, A. (2017). Word embeddings for improving rest services discoverability. En Monteverde, H. y Santos, R., editores, *CLEI*, pp. 1–8. IEEE.

[Mercado *et al.*, 2019] Mercado, V., Villagra, A., y Errecalde, M. L. (2019). Exploratory analysis of a new corpus for political alignment identification of argentinian journalists. En *XXV Congreso Argentino de Ciencias de la Computación (CACIC)(Universidad Nacional de Río Cuarto, Córdoba, 14 al 18 de octubre de 2019)*.

[Mikolov *et al.*, 2013a] Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov *et al.*, 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., y Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

[Pennington *et al.*, 2014] Pennington, J., Socher, R., y Manning, C. D. (2014). Glove: Global vectors for word representation. En *EMNLP*, volumen 14, pp. 1532–1543.

[Rangel, 2013] Rangel, F. (2013). Identifying information about gender, age, emotions and beyond. En *Proceedings of the 5th BCS IRSG Symposium on Future Directions in Information Access*.

[Stamatatos, 2009] Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society For Information Science and Technology*, 60(3):538–556.

[Weiss *et al.*, 2015] Weiss, D., Alberti, C., Collins, M., y Petrov, S. (2015). Structured training for neural network transition-based parsing. *CoRR*, abs/1506.06158.

[Zou *et al.*, 2013] Zou, W. Y., Socher, R., Cer, D., y Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. En *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1393–1398.