Enfoque híbrido para la correcta identificación de autores en bases de datos bibliográficas de libre acceso: el caso de *Google Scholar*

José Federico Medrano[1 [0000−0002−8779−029X]]

Facultad de Ingeniería - Universidad Nacional de Jujuy, Jujuy, Argentina
jfmedrano@fi.unju.edu.ar

Resumen En una Base de Datos Bibliográfica (BDB) un autor puede tener varios nombres y varios autores pueden compartir el mismo nombre simplemente debido a abreviaturas, nombres idénticos o errores ortográficos en las publicaciones. Esto puede producir ambigüedad en el nombre que puede afectar la atribución de créditos y cálculo de indicadores. La falta de normalización es muy común sobre todo en las BDB de libre acceso, *Google Scholar* (GS) es un claro ejemplo de ello. Aquí se presenta un enfoque para desambiguar los nombres de autor a partir de un conjunto de publicaciones provenientes de GS. Se propone un enfoque híbrido basado en redes de coautoría y reglas heurísticas para la detección de agrupaciones de autores más frecuentes. Los resultados preliminares evidencian la factibilidad del enfoque propuesto.

Palabras Clave: Desambiguación; Bibliometría; Google Scholar; Redes de Coautoría

1. Introducción

La calidad de cualquier estudio bibliométrico dependerá en gran medida de la base de datos a utilizar, *Scopus* y la *Web of Science (WoS)* han sido por décadas las BDB tradicionales en esta materia, principalmente por el enorme control de calidad asociado a ellas, sin embargo en los últimos años han empezado a perder popularidad frente a alternativas de libre acceso como GS [2,5]. Este cambio de paradigma vino dado por la enorme cobertura, facilidad de uso y gratuidad que ofrece GS frente a sus competidores.

Google Scholar no provee mecanismos de recolección de datos de forma automática. Otro de los problemas de GS está asociado con la calidad de los resultados, al tratarse de un indexador de documentos, los robots indexan todo el material científico-académico que consideran como tal de manera automática. Por ello existen registros mal formados, duplicados, mal agrupados (homónimos), particionados (parte del nombre de autor), campos incompletos, autores faltantes, entre los problemas más frecuentes. Aunque GS presenta ciertas limitaciones e inconsistencias, su amplia cobertura (superó los 389M de registros en enero de 2018 [1]) lo convierte en un buen candidato.

Muchos estudios han empleado una variedad de características de desambiguación tales como coautores, títulos de artículos, temas de artículos, correos-afiliaciones, etc., entre otros datos como lo expresa [8]. La literatura consultada al respecto da cuenta de las múltiples variantes del problema y de las diversas soluciones propuestas, utilizando métodos de aprendizaje supervisado [9], no supervisado [6] o en combinación de ambos esquemas [3]. Sin embargo, no existen trabajos que aborden el problema de la desambiguación empleando GS como origen de datos. Por ello la novedad de este trabajo que emplea la coautoría como mecanismo de agrupación, ya que la frecuencia en los autores conocidos representados por la coautoría, podrían discriminar las identidades de los autores con más claridad que otras características. Sumado a esto, y debido a la falta de normalización del conjunto de datos, se aplica un conjunto de reglas heurísticas para conformar nuevas agrupaciones o descartar registros incorrectos.

2. Metodología

Para resolver el problema planteado se diseñó un esquema híbrido que permite identificar en primer lugar las posibles variantes de nombres de autor comenzando con el nombre de un Autor Buscado (AB), mediante combinaciones y permutaciones de los nombres, apellidos e iniciales; y en segundo lugar identificando las redes de coautoría del autor. De este modo se descartan registros donde es casi imposible confirmar la autoría o relación con el autor buscado y entregando un conjunto de registros limpios y listo para ser analizado.

Se generó una aplicación web en C# que permite realizar búsquedas por autor en GS mediante consultas HTTP GET, el código HTML devuelto es procesado mediante expresiones regulares para delimitar tanto los registros resultantes como las partes de cada uno (título, nombre de autor, lugar de publicación, año de publicación, resumen y cantidad de citas). Es necesario aclarar que solo se cuenta con la información recolectada a partir de la consulta de autor, no se cuenta con ningún otro tipo de datos filiatorios del autor como lo realizado en [4].

En el Algoritmo 1 se ofrece en pseudocódigo el esquema propuesto. Cada registro del conjunto recolectado es analizado para comprobar si pertenece a AB, si el autor escribe en solitario se analiza si AB o alguna de las variantes del nombre es igual al autor del registro analizado. Si es así, la variante del nombre se almacena y el registro es agregado al *cluster* del mismo, caso contrario el registro se descarta. Cuando existe más de un autor, cada autor es comparado para calcular la similitud del nombre con AB. La función *CompararAutores(R.autor, AB)* utiliza la similitud *MongeElkan*[7], un esquema híbrido que se comporta muy bien para comparaciones basadas en pocos términos con mínimos errores. La función devuelve el autor (*NombreAutor*) y el mayor puntaje (*pMax*), si el puntaje obtenido supera un umbral definido (*UmbralNombre = 0.92*) quiere decir que *NombreAutor* es muy parecido a AB, por ello la función *ComprobarNombres()* empareja mediante una serie de reglas los nombres e iniciales para comprobar de que variante de nombre se trata. Seguido de esto se analizan los

Algorithm 1: Desambiguación de nombres

```
Data: AB Autor Buscado, ListaRegistros registros recolectados
Result: Conjuntos de registros según autores/coautores
foreach R en ListaRegistros do
    if R.size==1 then                                   /* Un único autor */
        if R.autor==AB or R.autor in Combinaciones(AB) then
            AgregarVarianteNombre(R.autor);
            AgregarRegistroCluster(R);
        end
    else                                                /* Más de un autor */
        pMax, NombreAutor ⟵ CompararAutores(R.autor, AB);
        if pMax > UmbralNombre then
            CoAutores ⟵ R.autor − NombreAutor;
            ComprobarNombres(R.autor, NombreAutor);
            AnalizarCoAutores(); DeterminarCluster();
        else
            ComprobarRegistro(R.titulo);
        end
    end
end
```

coautores (*AnalizarCoAutores()*) para identificar si alguno de ellos ya pertenece a una agrupación existente o se debe crear una nueva agrupación basada en los trabajos en común entre más de un coautor (*DeterminarCluster()*). Cuando $pMax$ no supera el umbral se realiza una consulta a la AK-API[1] con combinaciones del título normalizado (texto en minúsculas y sin signos de puntuación) de la publicación. Se analizan los primeros 10 resultados en busca del AB dentro de los autores de los resultados obtenidos. Esto se hace así porque hay casos en los que GS entrega los nombres de autor mal formados o particionados, si se logra encontrar el AB o alguna combinación del nombre, el registro es agregado al *cluster* del autor más productivo, caso contrario se descarta. El proceso finaliza entregando un listado de variantes de nombre de autor, tantas como agrupaciones distintas de coautores existan, junto a un conjunto de registros por cada variante, el usuario puede fusionar dichas agrupaciones o elegir las que desee para conformar un conjunto de publicaciones del autor buscado.

3. Conclusiones

El esquema propuesto se probó de forma efectiva en investigadores del campo de la Cienciometría, Bibliometría y Ciencias de la Información, sobre un total de 10 perfiles, extrayendo el corpus completo de registros de GS de cada autor (entre 95 registros el menos productivo y 700 el más productivo) y obteniendo

[1] https://www.microsoft.com/en-us/research/project/academic-knowledge/

resultados muy favorables. Por un lado se descartaron trabajos que no pertenecían a estos investigadores (entre un 5 % y 18 % de registros fueron eliminados de la lista de trabajos por no pertenecer al autor buscado), si bien la cantidad de registros se redujo producto de la limpieza, el conjunto resultante ofreció un número más cercano a la realidad y por ello el cálculo de indicadores estuvo libre de sesgos. Al menos con los investigadores analizados y sobre las pruebas realizadas, no se hallaron falsos positivos.

El aporte principal de este trabajo radica en proporcionar un esquema viable para el empleo de un motor académico de libre acceso como origen de datos para el análisis de la producción científico-académica de un investigador. Se comprobó que la extracción de datos es posible a pesar de no contar con mecanismos provistos por la propia base de datos. Utilizando fuentes de datos gratuitas y de amplia cobertura los indicadores mejoran notablemente frente a los indicadores calculados por las fuentes tradicionales, en este trabajo, los indicadores de productividad a partir de GS mejoraron entre un 32 % y 45 % frente a los calculados con *Scopus* o *WoS*.

Referencias

1. Gusenbauer, M.: Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases. Scientometrics **118**(1), 177–214 (2019)
2. Harzing, A.W., Alakangas, S.: Google scholar, scopus and the web of science: a longitudinal and cross-disciplinary comparison. Scientometrics **106**(2), 787–804 (2016)
3. Kim, J.: Evaluating author name disambiguation for digital libraries: a case of dblp. Scientometrics **116**(3), 1867–1886 (2018)
4. Liu, Y., Li, W., Huang, Z., Fang, Q.: A fast method based on multiple clustering for name disambiguation in bibliographic citations. Journal of the Association for Information Science & Technology **66**(3), 634–644 (2015)
5. Martín-Martín, A., Orduña-Malea, E., Thelwall, M., López-Cózar, E.D.: Google scholar, web of science, and scopus: A systematic comparison of citations in 252 subject categories. Journal of Informetrics **12**(4), 1160 – 1177 (2018)
6. Momeni, F., Mayr, P.: Evaluating co-authorship networks in author name disambiguation for common names. In: International Conference on Theory and Practice of Digital Libraries. pp. 386–391. Springer (2016)
7. Monge, A.E., Elkan, C.P.: The field matching problem: Algorithms and applications. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 267–270 (1996)
8. Tekles, A., Bornmann, L.: Author name disambiguation of bibliometric data: A comparison of several unsupervised approaches. arXiv preprint arXiv:1904.12746 (2019)
9. Zhang, B., Dundar, M., Al Hasan, M.: Bayesian non-exhaustive classification a case study: Online name disambiguation using temporal record streams. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 1341–1350 (2016)