ISSN 1870-4069

Detección automática de engaño en notas de opinión a partir de técnicas de perfilado de autores

Jonathan Serrano-Pérez[1], Javier Sánchez-Junquera[1], Hugo Jair Escalante-Balderas[1], Luis Villaseñor-Pineda[1,2]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, Laboratorio de Tecnologías del Lenguaje, Puebla, México

[2] Université d'Artois, Centre de Recherche en Linguistique Française, Arras, France

{js.perez, hugojair, villasen}@inaoep.mx, jjsjunquera@gmail.com

Resumen. El presente trabajo muestra los resultados alcanzados al aplicar un método originalmente propuesto para perfilado de autores a la detección automática de engaño. El método representa cada perfil (i.e. autores de cierto género o rango de edad) a través de *subperfiles* para cada categoría. Es decir, no supone que todos los jóvenes, o todos los adultos, escriben con el mismo estilo. Este trabajo, retoma la misma suposición, e intenta afinar la discriminación entre notas engañosas y verdaderas, al suponer que existe más de un posible estilo para redactar este tipo de textos. El presente trabajo analiza el comportamiento del método variando el número de subperfiles en dos tipos de colecciones usadas para la detección del engaño: reseñas sobre hoteles y temas controversiales. El método alcanzó resultados alentadores, difiriendo los resultados según el tipo de documentos donde se pretende detectar el engaño.

Palabras clave: minería de textos, clasificación no-temática de textos, detección de engaño.

Automatic Deception Detection in Opinion Notes using Author Profiling Techniques

Abstract. The present work shows the results achieved by applying a method originally proposed for author profiling to the automatic detection of deception. The method represents each profile (i.e. authors of a certain genre or age range) through *sub-profiles* for each category. That is, it does not assume that all young people, or all adults, write with the same style. This work takes up the same assumption, and attempts to refine the discrimination between deceptive and true notes, assuming

that there is more than one possible style to write such texts. This paper analyzes the behavior of the method by varying the number of sub-profiles in two types of collections used for the detection of deception: hotel reviews and controversial issues. The method achieved encouraging results, with results differing according to the type of documents where the deceit is intended to be detected.

Keywords: text mining, non-thematic text classification, deception detection.

1. Introducción

El uso de la internet se ha generalizado de tal forma que se recurre a este medio de información casi para cualquier cosa. En especial, consultar la internet para informarse sobre valoraciones de productos o servicios es habitual. De este modo, una persona que desea adquirir un producto o un servicio recurre a la web para responder preguntas como ¿el producto o servicio cumple con lo que promete?, ¿la tienda o el vendedor es confiable?, ¿la tienda o página web me ofrece alguna garantía por defectos o si no llega el producto?, entre otras. Básicamente, estas preguntas se responden con los comentarios o reseñas de compradores previos.

Algunos vendedores de productos y servicios, se han dado cuenta de esta situación y a través de estrategias poco éticas han intentado sacar provecho de este comportamiento al agregar comentarios positivos referentes a sus productos, y/o escribir reseñas negativas a productos o servicios de sus competidores.

La búsqueda de métodos automáticos para detectar opiniones que fueron escritas con la intención de engañar se le conoce como *detección de engaño*. La importancia de detectar automáticamente el engaño (u opiniones falsas[3]) es clara en situaciones como el caso de *TripAdvisor*. Dicho sitio cuenta con millones de opiniones de viajeros acerca de alojamientos, y se tiene particular interés en la detección de opiniones falsas, cuyo fin generalmente es aumentar o disminuir la reputación de un establecimiento por parte de propietarios o competidores, respectivamente[4]. Sin embargo, existen muchas otras situaciones en que este tipo de métodos podrían ser de ayuda al experto para la toma de decisiones, como es el caso de evaluación de veracidad de testimonios.

La detección automática del engaño recae principalmente en observar elementos que brinden evidencia de haber experimentado en *carne propia* los hechos relatados. Elementos como el uso de la primera persona, expresiones incluyendo valoraciones sensoriales, y la descripción puntual y detallada de la experiencia pueden proporcionar evidencia de la veracidad de la opinión. Trabajos previos han demostrado que estos elementos pueden capturarse a través de técnicas de

[3] Si bien las opiniones falsas no implican necesariamente la existencia de engaño, o sea, la intención de engañar, en este documento se mencionará "opiniones falsas" como expresión alternativa para referirse a "opiniones engañosas".

[4] https://www.tripadvisor.es/vpages/review_mod_fraud_detect.html

minería de texto, al capturar rasgos preponderantemente asociados al estilo de escritura de la reseña [10].

El presente trabajo explora y evalúa la aplicación de una técnica usada con éxito para la discriminación entre perfiles de autor identificando características como género o rango de edad. Dicha técnica [6] recurre al supuesto que el estilo y los temas tratados en el documento permiten discriminar, por ejemplo, a qué género pertenece el autor. Lo que es más, abre la posibilidad de capturar diversas actitudes entre autores pertenecientes a la misma clase, al considerar la existencia de *subperfiles*, ya que es difícil de imaginar que todos los autores recurren al mismo estilo de escritura. Extendiendo esta idea a la tarea de detección del engaño, no sólo deseamos discriminar entre opiniones verdaderas y falsas, sino incluso se puede suponer que el estilo en cada clase no es homogéneo. Es decir, es de suponer que existen diferentes estilos entre los autores de notas falsas así como diferentes estilos entre autores de notas verdaderas.

A continuación, en la Sec. 2, se presentan algunos trabajos relacionados a la detección de engaño. En la Sec. 3, se describe el método llevado a cabo. En la Sec. 4 se detallan los datos sobre los corpus de prueba y se muestran los resultados obtenidos junto con una breve discusión de los mismos. Finalmente se dan conclusiones preliminares en la Sec. 5.

2. Trabajo relacionado

Existen diferentes trabajos donde se intenta detectar las opiniones engañosas de las verdaderas. Estos trabajos difieren en las representaciones usadas así como en el tipo de documentos donde se desea hacer la detección. Respecto al tipo de documentos se identifican dos grandes tipos de colecciones: (i) opiniones *spam*[5] sobre productos o servicios, tales como libros, restaurantes, hoteles y doctores [10, 3, 9] y (ii) engaño en opiniones sobre tópicos controversiales como aborto, pena de muerte, y sentimientos sobre mejores amigos[8, 7, 11]. Las colecciones varían considerablemente no sólo por el tipo de contenido sino también desde el punto de vista psicológico. En la primera colección se recurrió a voluntarios para realizar el trabajo de redacción, y ellos estuvieron conscientes de que sus notas falsas no tendrían ninguna implicación. En el caso del otro tipo de colección, el autor estaba consciente de que plasmaba creencias propias y tendrían una repercusión sobre su imagen ante terceros, posibilitando así la presencia de emociones negativas vinculadas anteriormente con el acto de mentir [2, 16].

Respecto al tipo de atributos utilizados para enfrentar esta tarea, se han experimentado diferentes rasgos que se diferencian en cuanto a su complejidad y a lo que son capaces de capturar: n-gramas de palabras y de caracteres [15], estructuras sintácticas [3, 13], lista de criterios psicolingüísticos [5] y atributos semánticos [1].

[5] Opiniones *spam* o *fake reviews*, son opiniones engañosas, escritas de forma que parezcan auténticas, y en las que deliberadamente se da información falsa influyendo en la decisión de usuarios y clientes [2,4].

Sorprendentemente, las secuencias de n palabras (o n-gramas de palabras), han servido para discriminar engaño de no engaño con un desempeño superior al del ser humano. En [10] abordaron la tarea como una clasificación de textos mediante un clasificador basado en n-gramas de palabras. Con el propósito de modelar el contenido y el contexto, consideraron tres conjuntos de atributos: *unigramas*, la combinación de *unigramas* y *bigramas* ($bigramas^+$), y la combinación de *unigramas*, *bigramas* y *trigramas* ($trigramas^+$). Los autores contrastaron los resultados de la clasificación de 800 opiniones positivas sobre hoteles mediante n-gramas, frente a otro enfoque usando criterios psicolingüísticos ($LIWC$). Los resultados sugieren que con *unigramas* se discrimina mejor que con un conjunto de criterios preestablecidos como *LIWC*, y que aproximaciones sensibles al contexto ($bigramas^+$) pueden mejorar la clasificación (89 % de exactitud). En [9] se mostró igualmente una mayor efectividad de $bigramas^+$ (86 % de exactitud) frente a las predicciones de jueces humanos, esta vez incluyendo 800 opiniones negativas.

En el caso de opiniones controversiales, en [7] se recolectaron opiniones en tópicos de pena de muerte, aborto, y sentimientos hacia el mejor amigo. En este trabajo, se aplicó un proceso de *stemming*, eliminándose las diferentes variaciones de una misma palabra y tomándolas como sinónimos. El desempeño promedio de los tres dominios fue de un 70 % de exactitud. Como se mencionó en párrafos anteriores, se trata de colecciones de características muy distintas a las opiniones *spam*, de ahí la diferencia de resultados.

Finalmente, una representación más compleja que los simples patrones léxicos fue la empleada por [3] tratando de describir más ampliamente el estilo de los engañadores. Los autores aplicaron su método en opiniones de productos, servicios y opiniones controversiales. Este consistió en el uso de reglas de producción basadas en árboles de derivación de acuerdo a gramáticas libres de contexto (*CFG*, por siglas en inglés). Con esta información fue posible detectar engaño, alcanzando aún mejores resultados cuando estos atributos se combinan con atributos léxicos (90 % de exactitud). Claro está que este método depende de recursos lingüísticos incrementando su costo computacional y restringiendo su ámbito de utilidad.

Cabe mencionar un último trabajo que antecede y motiva el presente estudio. En [13] se probaron varias representaciones entre las que destacó el discriminar primeramente por género del autor para posteriormente detectar el engaño. Los autores indican en sus experimentos que los *unigramas* fueron la representación más robusta, que las mentiras en general son más difíciles de detectar que las verdades, y que las mentiras dichas por mujeres son más fáciles de identificar que las de los hombres.

El presente trabajo no distingue entre los géneros de los autores, información que no está presente en las colecciones de prueba. Pero sí se busca discriminar el engaño de la verdad afinando la granularidad de la representación al suponer que posiblemente existen *subperfiles* tanto entre los engañadores como entre aquellos que dicen la verdad. La siguiente sección describe la representación y metodología utilizadas.

3. Metodología

3.1. Representación de documentos

La representación usada parte directamente de lo expuesto en [6]. Esta técnica, también conocida como *SOA2*, hace la suposición que los autores de una clase de documentos están repartidos en diferentes *subperfiles*. Es por esto que se tiene interés en aplicar esta técnica en la detección del engaño, al suponer que podrían encontrarse diferentes *subperfiles* de mentirosos, lo cual podría aportar información significativa en el análisis y clasificación de nuevos documentos. Los siguientes párrafos detallan la aplicación del método *SOA2*.

Notación. Tomada de [6]:

- $D = \{(d_1, y_1), ..., (d_n, y_n)\}$, D es una colección de n parejas de documentos (d_i) y variables (y_i), donde la variable representa el perfil al cual está asociado el documento.
- $y_i \in P = \{p_1, ..., p_q\}$, P es el conjunto de diferentes perfiles y q es la cardinalidad de P.
- $V = \{v_1, ..., v_m\}$, V es el vocabulario de la colección de documentos D.
- v_i es representado como un vector $\mathbf{t_i} \in \mathbb{R}^q$, por lo tanto $\mathbf{t_i} = (t_{i,1}, ..., t_{i,q})$, donde cada elemento $t_{i,j}$ indica el grado de asociación entre el término v_i y el perfil p_j.
- $tf(d_k, v_i)$ es la frecuencia del término v_i en el documento d_k, $len(d_k)$ es la cantidad de términos que contiene el documento d_k.
- $\mathbf{x_k}$ representa al k-ésimo documento, donde $\mathbf{x_k} \in \mathbb{R}^q$.

Representación de función de los perfiles. Dado que se tiene una colección de documentos etiquetados, las diferentes etiquetas son vistas como los perfiles, por ejemplo, en el corpus de *OpSpam* se tienen las clases de engaño y veraz, por lo tanto se tendrían dos perfiles. Sabiendo esto, se llevará a cabo la representación del vocabulario del corpus, por lo que se crea una matriz del tamaño del vocabulario por la cantidad de perfiles (véase Figura 1). Como se puede observar en la Figura 1, cada perfil aporta información a las palabras que son usadas dentro del mismo, este aporte está dado por la ecuación 1. Posteriormente se hace un normalizado por perfil, es decir, se normalizan las columnas de la matriz, y finalmente se hace una normalización por término, es decir, por fila. De esta forma se construye la representación de los términos en el espacio de perfiles:

$$t_{i,j} = \sum_{\forall d_k : y_k == p_j} log_2(1 + \frac{tf(d_k, v_i)}{len(d_k)}). \quad (1)$$

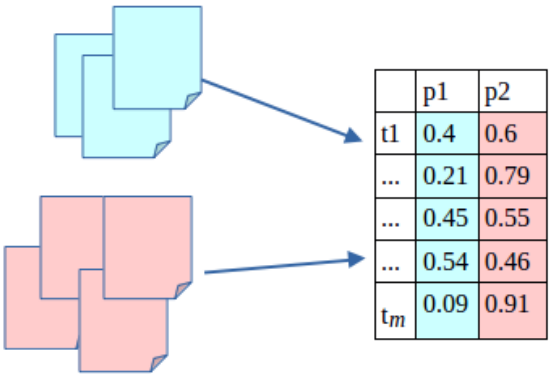

Fig. 1. Se crea una matriz de tamaño $m \times q$, donde m es el tamaño del vocabulario y q es el numero de perfiles, posteriormente los documentos de cada perfil aportan información a las palabras que son usadas dentro del mismo, este aporte esta dado por la ecuación 1. De esta forma se obtiene la representación de los términos en el espacio de perfiles.

Representación de documentos. Una vez que se tiene la representación de los términos en el espacio de perfiles, se usan estos para representar los documentos en el mismo espacio. Esto es, cada palabra del documento aporta información para la representación en el espacio de perfiles (véase Figura 2), este aporte está dado por la ecuación 2. Consecuentemente se obtienen las representaciones de los documentos en el espacio de perfiles:

$$\mathbf{x_k} = \sum_{v_i \in d_k} \frac{tf(d_k, v_i)}{len(d_k)} \times \mathbf{t_i}. \tag{2}$$

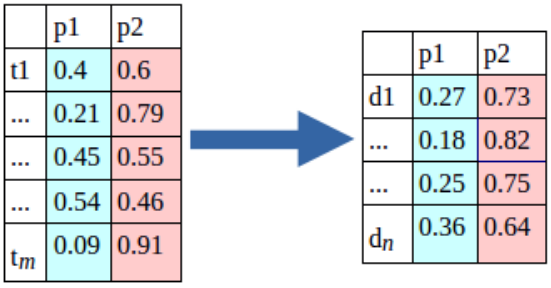

Fig. 2. Las palabras de cada documento, previamente representadas en el espacio de perfiles (tabla izquierda), son utilizadas para representar al documento en el mismo espacio de perfiles (tabla derecha), el aporte de cada palabra esta dado por la ecuación 2. De esta forma se obtiene la representación de los documentos en el espacio de perfiles.

Generación de subperfiles. Una vez que se tiene la representación de los documentos en el espacio de perfiles se procede a hacer la búsqueda de *subperfiles*,

lo cual se logra en dos etapas. Primeramente se agrupan los documentos que pertenecen al mismo perfil, y luego se aplica algún algoritmo de agrupamiento a cada perfil, por ejemplo *k-means*, véase Figura 3. El problema que se tiene al usar *k-means* (y otros algoritmos de agrupamiento), es que requiere la cantidad específica de agrupaciones que debe encontrar, sin embargo, muchas veces este valor es desconocido, por lo que una forma de obtener un buena cantidad de agrupaciones es ejecutar el algoritmo de agrupamiento varias veces, indicando diferentes cantidades en cada ocasión y usando una medida de validación para encontrar la mejor agrupación. En este trabajo se usó el algoritmo de *k-means* con el coeficiente de *Silhouette* para encontrar las mejores agrupaciones por perfil, usando las implementaciones de *scikit learn*.

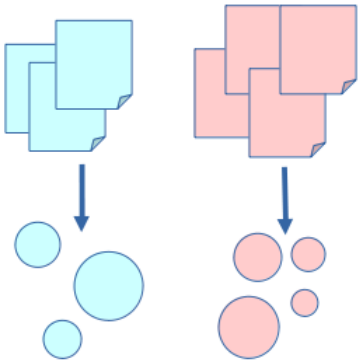

Fig. 3. Una vez representados los documentos en el espacio de perfiles, se aplican técnicas de agrupamiento para encontrar los subperfiles de cada perfil. Nótese que cada perfil puede tener diferente cantidad de subperfiles.

Los *subperfiles* encontrados (agrupaciones por perfil), serán vistos como las nuevas clases/perfiles, por ejemplo, en la Figura 3 se encontraron 7 *subperfiles*, de este modo se tendrían 7 clases/perfiles. El siguiente paso es repetir la *Representación de función de los perfiles* y posteriormente la *Representación de los Documentos*, por lo tanto, al final se obtiene una matriz de los documentos en el espacio de *subperfiles*, ver Figura 4.

	subp_1	subp_2	subp_3	subp_4	subp_5	subp_6	subp_7
d1	0.27	0.01	0.14	0.283	0.21	0.15	0.17
...	0.27	0.25	0.21	0.185	0.03	0.05	0.25
...	0.179	0.17	0.17	0.167	0.07	0.08	0.16
d_n	0.008	0.14	0.04	0.024	0.36	0.34	0.08

Fig. 4. Representación de documentos en el espacio de subperfiles.

Clasificación. Una vez finalizada la representación de los documentos a *SOA2*, la matriz resultante está lista para entrenar a cualquier clasificador que tome como entrada una matriz de ejemplos por atributos más una lista de la clase a la que pertenece cada ejemplo. Para este trabajo se han usado 2 clasificadores, el Naïve Bayes y una máquina de soporte vectorial (SVM), ambos clasificadores de WEKA 3.6, los resultados pueden verse en la Sec. 4.

Ventajas. Algunas de las ventajas de esta técnica son las siguientes:

- **Reducción de dimensión:** Comparado con una bolsa de palabras (BoW), el tamaño de los vectores en general es mucho mas pequeño que los vectores de la BoW, ya que el tamaño de los vectores de una BoW esta dado por el tamaño del vocabulario.
- **Matriz NO dispersa:** ligado con el punto anterior, y uno de los problemas bien conocidos de la BoW, es que se obtienen matrices con datos dispersos, sin embargo, esto no sucede con *SOA2*.

Desventajas. Una de las posibles desventajas de este método es la dificultad para determinar la cantidad óptima de *subperfiles* por clase, lo cual impacta en el rendimiento y eficacia del método.

3.2. Corpus y preprocesamiento

En este trabajo se han usado los siguientes cuatro corpus:

- ***OpSpam* [6, 5]:** Contiene 800 opiniones reales y 800 opiniones falsas acerca de hoteles situados en Chicago. Las opiniones verdaderas fueron extraídas de notas reales de *TripAdvisor*, mientras que las falsas fueron requeridas vía *Amazon Mechanical Turk* (AMT).
- **Temas controversiales (*Abortion, Death Penalty, Best Friend*) [4]:** En los 2 primeros temas, se pidió a algunas personas escribir su opinión (opiniones reales) y posteriormente se les pidió que escribieran una opinión contraria o lo que habían escrito previamente (opiniones falsas). De forma similar para el tercer tema, se pidió a algunas personas escribir sobre su mejor amigo (opiniones reales) y posteriormente se les pidió escribir sobre una persona que no soporten, como si fuera su mejor amigo (opinión falsa). Finalmente, se tienen 100 opiniones reales y 100 opiniones falsas por cada uno de los tres temas.

El preprocesamiento que se llevó a cabo en estos corpus fue reducción a minúsculas y la eliminación de signos de puntuación, enfocándose únicamente en las palabras. Para llevar a cabo la evaluación, se ha hecho con validación cruzada de 5 pliegues, es decir, 80 % para entrenamiento y 20 % para prueba por cada pliegue.

4. Resultados

En las tablas 1, 2, 3 y 4 se presentan los resultados para los corpus *OpSpam*, *Abortion*, *Best Friend* y *Death Penalty*, respectivamente. Dado que se generaron 5 pliegues, los resultados mostrados en las tablas para la Exactitud (E), Precisión (P), Recuerdo (R) y la medida F_1, son calculadas con el macro-promedio. La mejor exactitud la consigue en el corpus de *OpSpam*, alcanzando un 83.9, y el peor resultado lo consigue en el corpus de *Death Penalty*, donde solo logra alcanzar una exactitud del 56.0 %.

Tabla 1. Resultados en *OpSpam*.

	Máx.	Prom. Subperfiles			P		R		F_1	
Clasif.	Subperfiles	F	V	E	F	V	F	V	F	V
NB	2	2±(0)	2±(0)	**83.9**	0.855	0.825	0.819	0.860	**0.836**	**0.842**
SVM				81.9	0.879	0.776	0.740	0.897	0.802	0.832
NB	5	2±(0)	2±(0)	**83.9**	0.855	0.825	0.819	0.860	**0.836**	**0.842**
SVM				81.8	**0.880**	0.775	0.737	**0.898**	0.801	0.831
NB	10	3.6±(3.57)	3.6±(3.57)	83.8	0.850	**0.828**	**0.824**	0.852	**0.836**	0.840
SVM				80.5	0.870	0.762	0.717	0.892	0.785	0.821

Tabla 2. Resultados en *Abortion*.

	Máx.	Prom. Subperfiles			P		R		F_1	
Clasif.	Subperfiles	F	V	E	F	V	F	V	F	V
NB	2	2±(0)	2±(0)	74.0	0.797	0.707	0.660	0.820	0.718	0.756
SVM				74.5	**0.803**	0.710	0.660	**0.830**	0.721	0.763
NB	5	4±(1.41)	4±(1.41)	**76.0**	**0.803**	**0.733**	**0.700**	0.820	**0.745**	**0.771**
SVM				74.0	0.792	0.706	0.660	0.820	0.718	0.757
NB	10	6.8±(3.96)	5.8±(3.56)	73.0	0.782	0.696	0.650	0.810	0.708	0.748
SVM				74.0	0.780	0.719	0.690	0.790	0.727	0.748

En general, la cantidad de *subperfiles* de cada clase (i. e. F y V) tiene un comportamiento distinto entre las opiniones sobre hoteles y las controversiales. En particular, en las primeras el promedio de *subperfiles* es menor y con menor varianza, mientras que en las segundas parece aumentar proporcionalmente al parámetro de máximo agrupamiento, con excepción del corpus *Death Penalty*. Esto puede deberse tanto ala cantidad de datos que se tienen como a la forma en que cada corpus fue construido. Por ejemplo, para las opiniones verdaderas sobre hoteles, *TripAdvisor* insta a los usuarios a evaluar el hotel en función de aspectos específicos como localidad, limpieza, calidad del sueño, precios [2]. Sin embargo, las opiniones controversiales fueron adquiridas sin ningún tipo de restricción,

Tabla 3. Resultados de *Best Friend*.

	Máx.	Prom. Subperfiles			P		R		F_1	
Clasif.	Subperfiles	F	V	E	F	V	F	V	F	V
NB	2	2±(0)	2±(0)	78.0	0.833	0.768	0.740	0.820	0.771	0.779
SVM				**81.5**	0.899	0.772	0.720	0.910	0.791	**0.831**
NB	5	3.2±(0.44)	3.6±(0.89)	**81.5**	0.861	**0.793**	**0.760**	0.870	**0.800**	0.824
SVM				79.5	**0.911**	0.738	0.660	**0.930**	0.757	0.821
NB	10	8.2±(2.68)	6.8±(3.27)	80.5	0.871	0.774	0.730	0.880	0.784	0.819
SVM				78.0	0.858	0.735	0.670	0.890	0.747	0.803

Tabla 4. Resultados de *Death Penalty*.

	Máx.	Prom. Subperfiles			P		R		F_1	
Clasif.	Subperfiles	F	V	E	F	V	F	V	F	V
NB	2	2±(0)	2±(0)	55.5	0.557	0.556	**0.560**	0.550	**0.555**	0.550
SVM				55.0	0.562	0.543	0.46	0.64	0.502	0.585
NB	5	2±(0)	2±(0)	**56.0**	0.566	**0.557**	0.550	0.570	0.554	0.560
SVM				55.5	**0.570**	0.547	0.460	**0.650**	0.504	**0.591**
NB	10	3.4±(3.13)	2.8±(1.78)	44.5	0.557	0.536	0.504	0.580	0.512	0.551
SVM				54.5	0.554	0.540	0.450	0.640	0.492	0.582

por lo que los individuos tuvieron total libertad de expresarse en dichos temas pudiendo ser tan específicos o amplios según quisieran.

Para poner en contexto los resultados obtenidos, en la tabla 5 se comparan los resultados contra el método tradicional de bolsa de palabras (BoW), y otros métodos del estado del arte que no utilizan recursos externos o herramientas de análisis lingüístico.

Tabla 5. Comparación con el estado del arte

Corpus	**Trabajos**	**Exactitud**
OpSpam	*bigramas*$^+$ [1]	**86.0**
	SOA2	83.9
	BoW	84.5
Abortion	*unigramas* [3]	63.8
	SOA2	**76.0**
	BoW	69.5
Best Friend	*unigramas* [3]	74.5
	SOA2	**81.5**
	BoW	77.5
Death Penalty	*unigramas* [3]	58.1
	SOA2	56.0
	BoW	**62.5**

Como puede observarse los resultados obtenidos al aplicar *SOA2* son superiores en función del tipo de engaño. En el caso de *OpSpam* el método no muestra ventajas cayendo su rendimiento aún por debajo del método tradicional de *BoW*. En el caso de las controversiales, utilizar un enfoque basado en *subperfiles* tiene resultados alentadores, superando incluso el estado del arte en los corpus *Best Friend* y *Abortion*. Aún es necesario profundizar más en el análisis de estos resultados para determinar las diferencias de comportamiento entre los corpus de notas controversiales. No obstante, estos resultados sugieren que la hipótesis de que existen *subperfiles* de engañadores y veraces se debe tener en cuenta en futuros trabajos para la detección del engaño. Finalmente, los resultados de propuestas que emplean recursos externos [6] o reglas gramaticales [1] para la representación de los textos siguen estando más por encima de los alcanzados por los métodos simples. Así que en futuros trabajos se podría buscar estrategias que permitan incluir este tipo de información en las representaciones de *SOA2*, de forma que se considere esta información en la búsqueda de *subperfiles*.

5. Conclusiones y trabajo futuro

Se ha mostrado cómo el uso de la técnica *SOA2* [3], la cual crea una representación basada en *subperfiles*, ha alcanzado resultados interesantes, prueba de ello se puede apreciar en la tabla 5, donde se compararon los resultados con métodos del estado del arte. No obstante, el comportamiento difiere en función del tipo de engaño. En el caso de las valoraciones sobre hoteles el método no brinda ninguna ventaja, por el contrario con los corpus de notas controversiales se tienen resultados interesantes. Aún es necesario realizar un análisis más profundo para explicar este comportamiento el cual puede deberse tanto al tamaño de las colecciones, a la diversidad de tópicos y por ende al tamaño del vocabulario, etc.

Para el trabajo futuro se propone, por un lado: (i) retomar los subperfiles encontrados en el paso de Generación de Subperfiles, y agregarlos como subclases, lo cual daría un problema multidimensional, el cual podría ser tratado usando un enfoque de clasificadores encadenados, de modo que estas nuevas subclases aporten información relevante a la clase original. Alternativamente podría verse como un ensamble, donde se construye un clasificador para cada subclase, se evalúa el nuevo documento y la clase que tenga más votos será a la que pertenece (cada subclase pertenece en principio a una clase, por lo que el voto de la subclase es para la clase a la que pertenece); y por otro lado, (ii) dado que los métodos que usan recursos externos o información sintáctica han demostrado su potencial, se podría buscar estrategias que permitan incluir información de este tipo en la búsqueda de *subperfiles* dentro del *SOA2*.

Agradecimientos Este trabajo ha sido realizado con el apoyo del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través de las becas No. 634411 y No. 613411, y del proyecto CONACYT CB-2015-01-257383.

Referencias

1. Ott, M., Cardie, C., Hancock, J.T.: Negative deceptive opinion spam. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 497–501. Association for Computational Linguistics (2013), http://www.aclweb.org/anthology/N13-1053
2. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1. pp. 309–319. HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011), http://dl.acm.org/citation.cfm?id=2002472.2002512
3. Pérez-Rosas, V., Mihalcea, R.: Cross-cultural deception detection. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 440–445. Association for Computational Linguistics (2014), http://www.aclweb.org/anthology/P14-2072
4. Rosso, P., Cagnina, L.C.: Deception Detection and Opinion Spam, pp. 155–171. Springer International Publishing, Cham (2017), https://doi.org/10.1007/978-3-319-55394-8_8