UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departamento de Sistemas Informáticos y Computación

Detección de Opinion Spam usando PU-Learning

Doctorando

Donato Hernández Fusilier

Directores

Dr. Paolo Rosso

Dr. Manuel Montes y Gómez

Dr. Rafael Guzmán Cabrera

Enero 2016

Agradecimientos

Primero que nada le agradezco a Dios por permitirme llegar hasta este momento, el cual ya no tiene retorno. Es para mí un verdadero gusto el poder culminar la realización de los estudios de doctorado que empezaron hace ya mucho tiempo. Todo con la fe puesta en Dios y nada sin él.

Este trabajo no habría sido posible sin el apoyo incondicional e infinito de los Doctores Paolo Rosso, Manuel Montes y Gómez y Rafael Guzmán Cabrera, bajo cuya asesoría y supervisión se realizó este tema de tesis. Les agradezco el que siempre estuvieron al pendiente de mis resultados, que no permitieran que me perdiera en ningún momento y en particular por permitirme trabajar junto a ustedes, es invaluable el hecho de que me hayan dedicado su tiempo y experiencia durante el desarrollo de este trabajo, no hay palabras para agradecerles todo lo que me enseñaron.

También me gustaría agradecerle a la Universidad de Guanajuato, al Campus Irapuato-Salamanca y en particular a la División de Ingenierías por todas las facilidades brindadas para la realización de este trabajo.

Termino por agradecer a mi familia, a mi esposa Lizbeth, a mi hijo Sergio Eulalio y a mi hija Litzy, por su constante amor, confianza y apoyo a lo largo de los años que he durado en esta aventura. Estoy agradecido con mis hermanas Lupe y Ángeles, quienes junto con mis sobrinas Lucía y Claudia siempre estuvieron pendientes del trayecto que tuve para conseguir este trabajo. A mi padre Eulalio, que en paz descanse y que desde donde quiera que este se sienta orgulloso por este logro, y a mi madre Guadalupe, quien nunca dejo de alentarme para que siguiera adelante hasta terminar con esta meta en la vida.

Es a ellos y ellas a quienes dedico este trabajo.

Resumen

La detección de opiniones falsas o verdaderas acerca de un producto o servicio, se ha convertido en un problema muy relevante de nuestra época. Según estudios recientes hasta el 80 % de las personas han cambiado su decisión final basados en las opiniones revisadas en la web. Algunas de estas opiniones pueden ser falsas positivas, con la finalidad de promover un producto, o falsas negativas para desacreditarlo.

Para ayudar a resolver este problema se propone en esta tesis un nuevo método para la detección de opiniones falsas, llamado PU-Learning*. Este método aumenta la precisión mediante un algoritmo iterativo y resuelve el problema de la falta de opiniones etiquetadas.

Para el funcionamiento del método propuesto se utilizan un conjunto pequeño de opiniones etiquetadas como falsas y otro conjunto grande de opiniones no etiquetadas, del cual se extraen las opiniones faltantes y así lograr una clasificación de dos clases. Este tipo de escenario se ha convertido en una situación muy común en los corpus de opiniones disponibles.

Como una segunda contribución se propone una representación basada en n-gramas de caracteres. Esta representación tiene la ventaja de capturar tanto elementos de contenido como del estilo de escritura, permitiendo con ello mejorar la efectividad del método propuesto en la detección de opiniones falsas.

La evaluación experimental del método se llevó a cabo mediante tres experimentos de clasificación de opiniones utilizando dos colecciones diferentes. Los resultados obtenidos en cada experimento permiten ver la efectividad del método propuesto así como también las diferencias entre la utilización de varios tipos de atributos.

Dado que la falsedad o veracidad de las opiniones vertidas por los usuarios, se convierte en un parámetro muy importante en la toma de decisiones, el método

que aquí se presenta, puede ser utilizado en cualquier corpus donde se tengan las características mencionadas antes.

Resum

La detecció d'opinions falses o vertaderes al voltant d'un producte o servei s'ha convertit en un problema força rellevant de la nostra època. Segons estudis recents, fins el 80 % de les persones han canviat la seua decisió final en base a les opinions revisades en la web. Algunes d'aquestes opinions poden ser falses positives, amb la finalitat de promoure un producte, o falses negatives per tal de desacreditarlo.

Per a ajudar a resoldre aquest problema es proposa en aquesta tesi un nou mètode de detecció d'opinions falses, anomenat PU-Learning*. Aquest mètode augmenta la precisió mitjançant un algoritme iteratiu i resol el problema de la falta d'opinions etiquetades.

Per al funcionament del mètode proposat, s'utilitzen un conjunt reduït d'opinions etiquetades com a falses i un altre conjunt gran d'opinions no etiquetades, del qual se n'extrauen les opinions que faltaven i, així, aconseguir una classificació de dues classes. Aquest tipus d'escenari s'ha convertit en una situació molt comuna en els corpus d'opinions de què es disposa.

Com una segona contribució es proposa una representació basada en n-gramas de caràcters. Aquesta representació té l'avantatge de capturar tant elements de contingut com a d'estil d'escriptura, permetent amb això millorar l'efectivitat del mètode proposat en la detecció d'opinions falses.

L'avaluació experimental del mètode es va dur a terme mitjançant tres experiments de classificació d'opinions utilitzant dues coleccions diferents. Els resultats obtingut en cada experiment permeten veure l'efectivitat del mètode proposat, així com també les diferències entre la utilització de varis tipus d'atributs.

Ja que la falsedat o veracitat de les opinions vessades pels usuaris es converteix en un paràmetre molt important en la presa de decisions, el mètode que ací es

presenta pot ser utilitzat en qualsevol corpus on es troben les característiques abans esmentades.

Abstract

The detection of false or true opinions about a product or service has become nowadays a very important problem. Recent studies show that up to 80 % of people have changed their final decision on the basis of opinions checked on the web. Some of these opinions may be false, positive in order to promote a product/service or negative to discredit it.

To help solving this problem in this thesis is proposed a new method for detection of false opinions, called PU-Learning*, which increases the precision by an iterative algorithm. It also solves the problem of lack of labeled opinions.

To operate the method proposed only a small set of opinions labeled as positive and another large set of opinions unlabeled are needed. From this last set, missing negative opinions are extracted and used to achieve a two classes binary classification. This scenario has become a very common situation in the available corpora.

As a second contribution, we propose a representation based on n-grams of characters. This representation has the advantage of capturing both the content and the writing style, allowing for improving the effectiveness of the proposed method for the detection of false opinions.

The experimental evaluation of the method was carried out by conducting three experiments classification of opinions, using two different collections. The results obtained in each experiment allow seeing the effectiveness of proposed method as well as differences between the use of several types of attributes.

Because the veracity or falsity of the reviews expressed by users becomes a very important parameter in decision making, the method presented here, can be used in any corpus where you have the above characteristics.

Índice general

Índice de figuras

Lista de algoritmos

Índice de tablas

Capítulo 1

Introducción

En este capítulo se presenta la introducción al tema de las opiniones falsas y los métodos empleados para su detección. En el apartado 1.1 se presenta la descripción de las opiniones falsas. En el apartado 1.2 se describen algunos de los trabajos realizados para la detección de las opiniones falsas. En el apartado 1.3 se describen los objetivos general y particulares que debe cumplir este trabajo de tesis. En el apartado 1.4 se describen los antecedentes de los métodos de aprendizaje utilizados en la detección de opiniones falsas. En el apartado 1.5 se describe el nuevo método para la detección de opiniones falsas. En el apartado 1.6 se describe de una manera resumida algunos de los resultados alcanzados en los experimentos realizados, considerando el nuevo método así como su combinación con la representación basada en n-gramas de caracteres. Por último, en el apartado 1.7 se presenta la organización de la tesis.

En la época actual por medio de las comunicaciones digitales es posible adquirir casi cualquier producto y contratar toda clase de servicios, sin haber tenido que cruzar una sola palabra con persona alguna. Por otro lado la información generada por los usuarios adquiere un valor muy significativo, tanto para los consumidores como para los fabricantes. En consecuencia para poder organizar y filtrar todo este tipo de información son necesarias nuevas herramientas, que nos permitan tomar la mejor decisión respecto a la adquisición o el rechazo de estos productos o servicios.

Con el uso general de las tecnologías de comunicación es cada vez más común que los usuarios de servicios y los consumidores de productos escriban sus opiniones a favor o en contra de lo que adquirieron. Estas referencias escritas comúnmente en foros, blogs y en general en las redes sociales, sirven de ayuda a otros consumidores que desean adquirir productos o servicios similares. También sirven a los fabricantes o prestadores de servicios para identificar nuevas áreas de oportunidad por parte de los consumidores y les permite saber no solo la opinión sobre los mismos, sino además ver sus usos, costumbres, satisfacción, etc.

Todo esto conlleva a un gran problema denominado *Opinion spam*, que en otras palabras, son opiniones falsas, escritas deliberadamente para promover o desacreditar un producto o servicio. Son opiniones escritas por personas que no han adquirido producto o servicio alguno, pero que fueron contratados para escribir opiniones engañosas (Fitzpatrick et al., 2015).

Estas opiniones falsas hacen creer a los posibles usuarios, que el producto o servicio es muy bueno o muy malo, según sea la causa para la cual fueron inducidos. Los consumidores utilizan las opiniones para recibir información sobre los productos, tales como calidad y utilidad. También son utilizadas para proporcionar datos sobre su propia experiencia con el producto a otros consumidores. Por otro lado, los fabricantes utilizan estos comentarios para identificar características que son importantes para los consumidores. Estas características son entonces incluidas en la comercialización y desarrollo de nuevos productos o servicios.

Por estas razones los proveedores de productos o servicios están interesados en poder detectar de una manera automática y eficiente las opiniones falsas. Con la finalidad de que solo existan opiniones verdaderas y lograr que los posibles usuarios obtengan una impresión real del producto o servicio.

Este trabajo de investigación se orienta principalmente hacia la detección de opiniones falsas mediante un nuevo método que está basado en el método original de PU-Learning. El cual tiene entre sus características principales la de aprender a partir de ejemplos positivos y requerir solamente una pequeña cantidad de datos etiquetados, junto con un conjunto grande de datos no-etiquetados para su entrenamiento. En el siguiente apartado se introducen algunos conceptos acerca de las características de las opiniones falsas.

1.1. Opiniones falsas

Las opiniones falsas, también llamadas *Opinion spam*, se comportan diferente a otros tipos de spam que existen. Entre los más comunes tenemos: el del correo electrónico (e-mail spam) (Graham, 2004) y el de las redes sociales como son Twitter, Facebook, etc. (Song et al., 2011).

Las opiniones falsas pueden llegar a ser muy dañinas ya que según estudios recientes (Feng et al., 2012a,b; Ott et al., 2011), ejercen una influencia de hasta el 80 % en la decisión final del usuario para adquirir o rechazar un producto o servicio. De acuerdo con la página "How online reviews affect your business"[1] es necesario tener herramientas adecuadas para detectar este tipo de opiniones falsas y en su caso hasta eliminarlas del conjunto de opiniones. De esta manera se disminuye el efecto que tienen en la toma de una decisión de compra o de rechazo por parte de los posibles usuarios de un producto o servicio.

El problema de detectar las opiniones falsas es muy complejo, incluso para las personas. Para darnos una idea de la complejidad de esta tarea, tratemos de distinguir cuál de las opiniones que se presentan a continuación es verdadera[2].

[1] http://mwpartners.com/positive-online-reviews. Visited: April 2, 2014.

[2]Estas opiniones aparecen en su idioma original inglés, para no alterar su contenido al traducirlas

Opinión 1

Located right in the heart of downtown Chicago i visited this hotel on a business trip. The first things i noticed was how friendly the staff is, and how clean the rooms are (even compared to other luxury hotels) seeing as i had a night layover before my meeting i decided to stop by the Terrace and get a drink, i got a mojito and i must say i was very impressed, i've had my share of mojitos and it was one of the better ones, once i retired back to my room i found the sound system to be excellent, bed was soft and the TV didn't appear burned out at all. Overall i give this a five out of five, the staff was helpful and friendly the whole time and the atmosphere was perfect, i wish i could have stayed longer.

Opinión 2

I just left the Conrad Chicago and have nothing but good things to say. We used Hilton points to stay there but paid $35/night to upgrade to a junior suite. We were in room 1519. I would recommend people ask for an odd numbered room so that they will have a view looking north on Michigan Ave. Our room was very modern with 2 large HD/flat panel tv and Bose 3 speaker sound system. 4 big windows that actually open. The bed lineds were very nice, the towels were OK. The room was clean and the bath room had a seperate shower and tub. Marble everywhere. Robes, slippers, etc. The work out room was good with different machines and some free weights. There are clean towels in the workout room too. We never ate in the hotel so I can't comment on the food or the prices. There is a Starbucks across the street (Michigan) and an Espresso bar attached to the Nordstrom that is on the ground floor of the hotel. It was very nice to have the little mall right under the hotel. The location of the hotel could not be better. We were walking distance to everything. Every kind of shopping or dinning experience you could possibly want was within 2 miles. Over all the hotel had a boutique feel to it even though it is part of the hilton chain.

Como se puede observar a simple vista parece difícil distinguir porque una opinión es falsa (la primera) y la otra es verdadera. Observando con más detalle el estilo de la escritura empleada en cada una de las opiniones podemos notar los siguientes puntos:

a) En la opinión falsa se omiten los detalles, se habla muy en general de lo que debe incluir una habitación de cualquier hotel.

b) En la opinión verdadera se tienen más detalles acerca de la habitación.

c) En la opinión verdadera aparece la localización de los objetos con más detalle.

d) En la opinión verdadera se menciona la cercanía del hotel a otros lugares como centros comerciales o restaurantes.

e) La cantidad específica de ciertos parámetros como los costos, las televisiones y las ventanas. Solo están presentes en una opinión real.

Es evidente que la persona que escribió la segunda opinión si estuvo hospedada en el Hotel Conrad, ya que la descripción detallada que realiza sobre la habitación, solamente la puede hacer alguien que si estuvo hospedado ahí.

Las opiniones de texto que existen sobre los productos o servicios, pueden ser clasificadas usando métodos automáticos de categorización de textos. Algunas de estas categorías pueden ser de acuerdo a una característica particular de las opiniones, como por ejemplo:

- *Sentimiento.* En este caso se clasifican de acuerdo a si son a favor del producto o servicio*(favorables)* o si son en contra*(desfavorables)*.
- *Veracidad.* Esta clasificación se realiza en base a si son opiniones reales*(verdaderas)* o engañosas*(falsas o spam)*. Este tipo de clasificación es la que más nos interesa detectar, dada la influencia que tiene en la decisión final de compra o rechazo del consumidor.
- *Dominio.* En esta clasificación lo que se trata es buscar si la opinión pertenece a un dominio específico.

En el siguiente apartado se presenta una descripción de las características de los métodos que se utilizan en la detección de las opiniones falsas.

1.2. Detección de las opiniones falsas

Una de las maneras tradicionales de realizar la detección de las opiniones falsas, es por medio de varios profesionales entrenados que emiten su veredicto acerca de

la opinión en cuestión. Después se toma la decisión final en base a la opinión de la mayoría de los profesionales. Se ha demostrado, según (Ott et al., 2011, 2013), que la precisión que se obtiene cuando se realiza la detección de las opiniones falsas por medio de humanos, es de alrededor del 60 %. Este valor está apenas arriba del valor que se obtiene eligiendo al azar*(50 %)*. Cuando el conjunto de opiniones a revisar es muy grande, esta forma de detectar las opiniones falsas, se vuelve incosteable y muy lenta. Además se disminuye la precisión a valores que pueden estar incluso por debajo del azar.

El problema de la detección de las opiniones falsas se ha vuelto muy interesante. Algunas aproximaciones han demostrado ser eficaces en otras tareas semejantes y en la actualidad existen varios métodos que usan la categorización de texto. Dentro de estos métodos podemos mencionar el trabajo de (Drucker et al., 2002) con máquinas de vectores de soporte *(SVM)* y el de (Jindal et al., 2010) basado en reglas. La precisión lograda empleando este tipo de métodos, es de alrededor del 80 %, pero la desventaja de este tipo de métodos es que requieren un número grande de opiniones manualmente etiquetadas para su buen funcionamiento.

Los problemas reales que se presentan en la detección de opiniones falsas son los siguientes:

- En la mayoría de las ocasiones no se dispone de suficientes ejemplos etiquetados de ambas clases (falsas y verdaderas) para el entrenamiento.
- La mayoría de los métodos actuales para la detección de las opiniones falsas solamente consideran la información del contenido y no del estilo con el cual fueron escritas.

Con respecto al problema de la información del contenido podemos agregar que en algunas ocasiones se dispone además de otro tipo de información del emisor, información temporal (fecha, hora y lugar) y cantidad de opiniones emitidas por un usuario, etc. Con los atributos de representación adecuados, la información del estilo con el cual se escribieron las opiniones resulta muy valiosa. Se puede obtener una mejor precisión en la detección de las opiniones falsas si se emplean atributos que sean capaces de capturar tanto el contenido como el estilo de escritura de las opiniones falsas.

Muy a menudo algunos atributos se presentan con muy baja frecuencia en las opiniones falsas. Estos atributos son en parte redundantes, mas sin embargo generalmente son incluidos, dada la escasez de opiniones etiquetadas en los dominios de los experimentos.

El presente trabajo de investigación está enfocado en superar las dificultades de la escasez de datos etiquetados y de la captura de información del estilo. Para resolver estos problemas se propone un nuevo método basado en el método de PU-Learning (Positives Unlabeled-Learning). También se propone la aplicación de los n-gramas de caracteres como atributos de representación, para capturar tanto el contenido como el estilo de escritura de las opiniones bajo estudio.

En el siguiente apartado se presentan los objetivos que debe cumplir el presente trabajo de tesis.

1.3. Objetivos

El objetivo general de este trabajo de tesis es el siguiente:

- *Desarrollar un método para la detección de opiniones falsas, que considere tanto características temáticas como estilísticas de los textos y que además sea adecuado para su aplicación en escenarios reales que presentan una escasez de datos etiquetados.*

Los objetivos particulares para el método son los siguientes:

- *Diseñar un método para la detección de opiniones falsas basado en PU-Learning.*
- *Proponer una representación de las opiniones con atributos que capturen tanto el contenido temático como el estilo de escritura de los documentos.*
- *Evaluar el método con opiniones de diferentes dominios para comprobar su efectividad en ambientes de dominios cruzados.*

En el siguiente apartado se establecerán algunos conceptos relevantes para el entendimiento del método .

1.4. Antecedentes

Los métodos de aprendizaje han sido ampliamente usados para la clasificación de documentos de texto en general. Se han utilizado diferentes algoritmos de aprendizaje automático como por ejemplo: Naïve Bayes (NB) (Lewis, 1998) y máquinas de vectores de soporte (SVM) (Cortes and Vapnik, 1995). Junto con estos algoritmos se han empleado diferentes tipos de atributos como son los n-gramas de palabras unigramas, bigramas y trigramas. También se han utilizado las combinaciones de estos atributos, tratando siempre de capturar la mayor cantidad de información acerca del documento.

En algunos casos se han empleado atributos que buscan capturar información acerca del estilo de escritura de los documentos de texto. En algunos trabajos se ha visto que se puede emplear por ejemplo: n-gramas de palabras (Ott et al., 2011), n-gramas de caracteres (Blamey et al., 2012), LIWC (Language Inquire and Word Count, que consiste en analizar el tipo de palabras empleadas y determinar a cual categoría pertenecen dentro de cada oración) (Ott et al., 2013), y otras clases de atributos.

La detección de las opiniones falsas se ha realizado empleando los clasificadores de texto y la selección de los atributos adecuados contribuye a mejorar el rendimiento de los clasificadores. Se han hecho experimentos usando diversos tipos de atributos como son los unigramas (Ott et al., 2011), bigramas (Ott et al., 2013), adjetivos para la detección del sentimiento (Feng et al., 2012b), etc.

En la mayoría de los experimentos realizados en la detección de opiniones falsas, se ha tomado como base una clasificación de dos clases *(binaria)*. Este tipo de clasificadores requieren un número grande de instancias etiquetadas en ambas clases para su buen funcionamiento. Esta condición es muy difícil de superar en muchos de los corpus de opiniones que se tienen disponibles actualmente. Es por esto que se propone resolver este problema utilizando los clasificadores de una sola clase, cuyas características se describen en el siguiente sub-apartado.

1.4.1. Clasificadores de una sola clase

Los clasificadores de una sola clase se distinguen de los clasificadores tradicionales, en que solo está disponible la información de una clase, la clase objetivo, también llamada positiva. Este tema es tan amplio que podríamos escribir mucho más al respecto (Tax, 2001).

Por ejemplo en el tema de atribución de autoría, distintos autores se han adjudicado la misma obra (Binongo, 2003). Una manera de identificar al autor verdadero sería la de analizar el estilo de la escritura en varias de las obras escritas por un mismo autor. Esta información pasaría a formar parte de la única clase existente *(la positiva)*. La otra clase *(la no-etiquetada)* estaría formada por todas aquellas obras que no fueron escritas por el autor mencionado. La clase no-etiquetada se convierte entonces en una colección tan grande que sería imposible tratar de modelarla, lo que es más desalentador aún, podríamos decir que es infinita.

En esta clase de problemas es donde se deben emplear los clasificadores de una sola clase. Dentro de los clasificadores de una sola clase podemos mencionar: los que están basados en métodos de densidad probabilística; Gaussianos y estimación de Parzen, los que están basados en métodos de contorno; K-centroides y vecinos cercanos (Tax and Duin, 2001). En estos métodos se trata de determinar si un elemento pertenece o no a la clase positiva por medio del cálculo de una medida, ya sea de densidad o de distancia.

Los clasificadores de una sola clase, son la base para el método de PU-Learning que se describe en el siguiente sub-apartado.

1.4.2. El método de PU-Learning

En el método de PU-Learning se tiene información de una sola clase, la clase positiva *(en nuestro caso la clase de las opiniones falsas)*, la cual generalmente es pequeña. Además se cuenta con otra clase *(la clase no-etiquetada)*, que contiene un gran número de instancias. A diferencia de los clasificadores de una sola clase, la clase no etiquetada es grande, pero sigue siendo una colección finita. Es posible modelarla y representa un conjunto de datos no etiquetados *(untagged)* que se compone de

instancias de las dos clases*(falsas y verdaderas)*.

El método de PU-Learning usa una estrategia de dos pasos para construir un clasificador de dos clases (Liu et al., 2002, 2003; Zhang and Zuo, 2009), siendo estos los siguientes:

a) Identificar de manera automática un conjunto de opiniones verdaderas confiables *(reliable negatives)*. Este conjunto se forma a partir de las instancias extraídas del conjunto no etiquetado.

b) Usar un algoritmo de aprendizaje con el conjunto de entrenamiento refinado, para construir un clasificador de dos clases.

En el algoritmo 1, se muestra la descripción formal del método. En este algoritmo P es el conjunto de ejemplos positivos (opiniones falsas) y U_i representa el conjunto de ejemplos no- etiquetados *(untagged)* en la iteración i. U_1 es el conjunto no etiquetado original. C_i se utiliza para representar el clasificador que se construyó en la iteración i, y Q_i indica el conjunto de casos clasificados como opiniones verdaderas por el clasificador C_i. Estas instancias se van extrayendo del conjunto no etiquetado y se añaden al conjunto de opiniones verdaderas confiables *(Reliable Negatives)* RN_i, para la siguiente iteración. La línea 9 del algoritmo muestra el criterio de paro que tiene que ser alcanzado para detener el ciclo, $|Q_{i-1}| > \emptyset$. La idea de este criterio de paro es permitir una continua pero gradual adición de los casos etiquetados como opiniones verdaderas por el clasificador C_i, desde el conjunto U_i al conjunto de opiniones verdaderas confiables RN_i, hasta que ya no existan más instancias que agregar. El método devuelve entonces un clasificador C_{i-1}, formado a partir de las clases P *y* RN_{i-1}, el cual se puede emplear para el conjunto de prueba.

Siguiendo estos pasos se obtiene un clasificador de dos clases. Este clasificador es capaz de proporcionar una mejoría de alrededor del 30 %, en la precisión comparada con la obtenida al haber usado un clasificador de una sola clase.

En el siguiente apartado se presenta el método de PU-Learning* que es una solución basada en el método de PU-Learning. En esté nuevo método se obtiene de una manera diferente la clase ausente, a como se realiza en el método de PU-Learning.

Algoritmo 1 El método de PU-Learning

1: $i \leftarrow 1$
2: $P \leftarrow$ `ejemplos positivos (opiniones falsas)`
3: $U_i \leftarrow$ `ejemplos no-etiquetados`
4: $C_i \leftarrow Generar\_Clasificador\ (P, U_i)$
5: $Q_i \leftarrow Extraer\_Verdaderas\ (C_i(U_i))$
6: $i \leftarrow i + 1$
7: $RN_i \leftarrow Q_{i-1}$
8: $U_i \leftarrow U_{i-1} - Q_{i-1}$
9: **mientras** $|Q_{i-1}| > \emptyset$ **hacer**
10: $\quad C_i \leftarrow Generar\_Clasificador\ (P, RN_i)$
11: $\quad Q_i \leftarrow Extraer\_Verdaderas\ (C_i(U_i))$
12: $\quad i \leftarrow i + 1$
13: $\quad RN_i \leftarrow RN_{i-1} + Q_{i-1}$
14: $\quad U_i \leftarrow U_{i-1} - Q_{i-1}$
15: **devolver** Clasificador C_{i-1}

1.5. El método de PU-Learning*

El método propuesto, al cual llamaremos PU-Learning* de aquí en adelante, consiste en una modificación del método de PU-Learning, en la manera como se obtiene la otra clase *(la clase de las opiniones verdaderas)* que no está presente. En el método de PU-Learning original esta clase se obtiene del conjunto no-etiquetado y se le van agregando más instancias en cada iteración.

En el método de PU-Learning*, se van retirando las instancias que se clasifican como opiniones falsas de la clase no-etiquetada, es decir se va disminuyendo el tamaño y la cantidad de opiniones falsas presentes en el conjunto no-etiquetado. En cada iteración se busca que las instancias que permanezcan en la clase no-etiquetada, sean solamente aquellas instancias que pertenecen a la clase que no cuenta con ejemplos etiquetados, opiniones verdaderas, desde un inicio.

Esta manera gradual de obtener la clase que contiene las opiniones verdaderas confiables, presenta la ventaja con respecto al método de PU-Learning de conservar en el conjunto no-etiquetado, aquellas instancias que no se parecen a las opiniones falsas. Este tipo de información es la única información confiable que se tiene y de la cual parte el método.

Al igual que el método de PU-Learning esta operación se realiza en dos pasos. De una manera iterativa se va depurando cada vez el conjunto no etiquetado, hasta llegar a un límite determinado por una condición de paro. Cuando esta condición se cumple, el proceso iterativo se detiene y se obtiene como resultado un clasificador de dos clases.

El método de PU-Learning* funciona en cualquier dominio, trabaja con pocas opiniones etiquetadas y también es independiente del tipo de atributos que se utilice. Este tipo de restricciones se presentan muy a menudo en los corpus de opiniones que se tienen disponibles en la web, por ejemplo el corpus de opiniones sobre hoteles de Chicago[3]. Este corpus está completamente etiquetado y nos servirá para realizar las pruebas que permitan medir la efectividad del método.

El método de PU-Learning* está representado en el algoritmo 2. En este algoritmo P es el conjunto de opiniones falsas y U_i representa el conjunto de datos no etiquetados en la iteración i. En las lineas 2 a 5 se establecen los valores iniciales de los conjuntos para asegurar que se repita al menos una vez el ciclo. C_i se utiliza para representar el clasificador que se construyó en la iteración i, W_i indica el conjunto de instancias clasificadas como falsas por el clasificador C_i. Estas instancias se van extrayendo del conjunto no etiquetado y se desechan en cada iteración. Por lo tanto, la clase no-etiquetada para la siguiente iteración se define como $U_i = U_{i-1} - W_i$.

La línea 6 del algoritmo muestra el criterio de paro que tiene que ser alcanzado para detener el ciclo, $|W_i| <= |W_{i-1}|$ y $|P| <= |U_i|$. La idea de este criterio es permitir una continua pero gradual reducción del conjunto U_i, extrayendo en cada iteración las opiniones etiquetados como falsas por el clasificador C_i, desde el conjunto U_i.

En la figura 1.1 se presenta el diagrama a bloques del método de PU-Learning*. El método parte de los conjuntos P (conjunto de opiniones falsas) y U_i (conjunto de ejemplos no etiquetados). Después se extraen del conjunto U_i las opiniones que el clasificador C_i considera como falsas para formar el conjunto W_i (conjunto de opiniones clasificadas como falsas en la iteración i). Estas acciones corresponden al primer paso del método. Con estos dos nuevos conjuntos, P y $U_i = U_{i-1} - W_i$, se

[3] http://myleott.com/op_spam

Algoritmo 2 El método de PU-Learning*

1: $i \leftarrow 1$
2: $P \leftarrow$ `ejemplos positivos (opiniones falsas)`
3: $U_i \leftarrow$ `ejemplos no-etiquetados`
4: $W_i \leftarrow U_i$
5: $W_{i-1} \leftarrow U_i$
6: **mientras** ($|W_i| <= |W_{i-1}|$ and $|P| <= |U_i|$) **hacer**
7: $\quad C_i \leftarrow Generar\_Clasificador\ (P, U_i)$
8: $\quad i \leftarrow i + 1$
9: $\quad W_i \leftarrow Extraer\_falsas\ (C_{i-1}(U_{i-1}))$
10: $\quad U_i \leftarrow U_{i-1} - W_i$
11: **devolver** Clasificador C_{i-1}

vuelve a construir un clasificador y se clasifica nuevamente el conjunto reducido U_i. Esta operación se repite mientras no se alcancen las dos condiciones del criterio de paro, indicando que ya no existe otra instancia que el clasificador etiquete como falsa y que deba ser extraída del conjunto U_i. Por último se devuelve un clasificador de dos clases C_{i-1} formado a partir de los conjuntos P y U_{i-1}.

En el siguiente apartado se presentan alguno de los resultados obtenidos en la aplicación de los dos métodos descritos en los sub-apartados anteriores.

1.6. Metodología

En este apartado se presenta un resumen de la manera como se han estructurado los experimentos para evaluar los métodos descritos en la sección anterior. En el subapartado 1.6.1 se describe de una manera breve el método de PU-Learning. En el siguiente subapartado 1.6.2 se describe una primer variante del método de PU-Learning y después en el subapartado 1.6.3 se presenta el método de PU-Learning* .

También se describen de una manera breve algunos de los resultados correspondientes a la aplicación de los métodos, en la detección de las opiniones falsas, utilizando como atributos los n-gramas de palabras así como también los n-gramas de caracteres. En el subapartado 1.6.4 se describe la implementación del método de

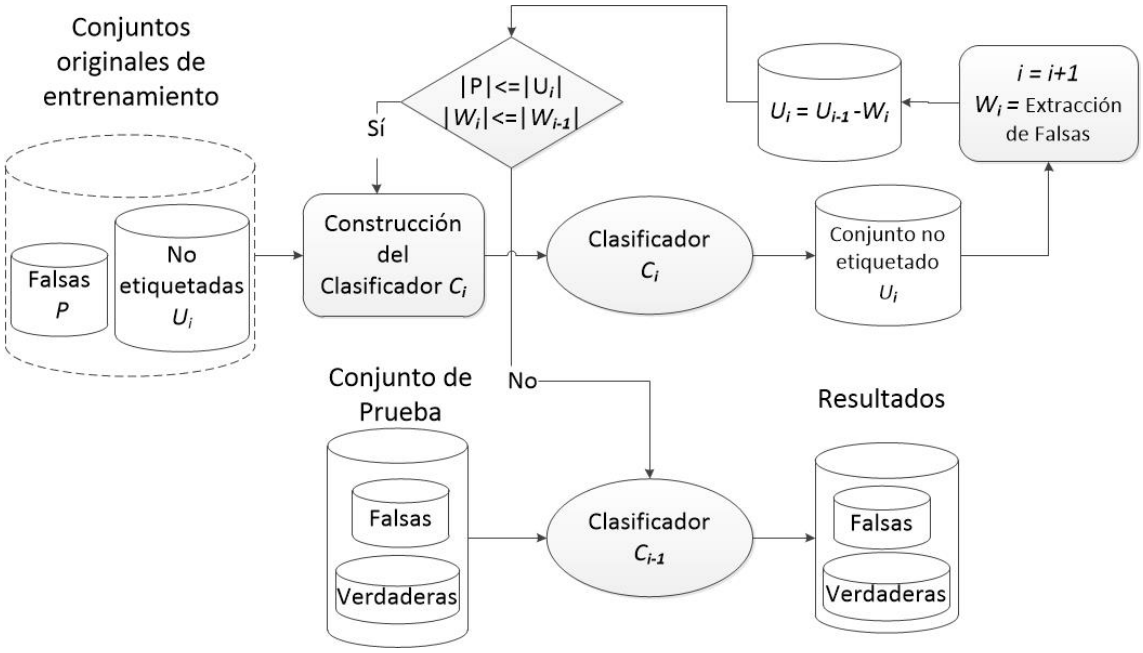

Figura 1.1: Construcción del clasificador con el método PU-Learning*.

PU-Learning* en ambientes de dominios cruzados, donde se entrena con el dominio de hoteles y se prueba con otros dos dominios diferentes (restaurantes y médicos).

1.6.1. Método PU-Learning

Este método tiene como ventaja principal que solo requiere de un conjunto de instancias etiquetadas (opiniones falsas) y logra extraer las instancias de la otra clase (opiniones verdaderas) de una manera iterativa, a partir de un conjunto de opiniones no-etiquetadas. Las instancias que se extraen del conjunto no-etiquetado, se van almacenando en un conjunto llamado *opiniones verdaderas confiables*.

La desventaja principal del método es que generalmente los conjuntos que se obtienen tras su aplicación, están desbalanceados y la mejora que se obtiene en la detección de opiniones falsas no es muy significativa. Esto es debido a que generalmente predomina el conjunto de instancias que fueron extraídas y que representa la clase de las opiniones verdaderas.

Este método se aplicó a la tarea de detección de las opiniones falsas, se utilizó un

conjunto pequeño de opiniones etiquetados *(20, 40, 60, 80, 100 y 120 opiniones falsas)*. En este caso fueron las opiniones favorables acerca de algúnos hoteles de Chicago. Además se usó un conjunto grande de opiniones*(520 opiniones)*, que no estaban etiquetadas del cual se extrajó el conjunto de opiniones verdaderas confiables.

Cuando se evaluó el conjunto de prueba con el clasificador obtenido mediante la aplicación del método de PU-Learning, se alcanzó un f-measure promedio de 0.690. Estos resultados aparecen publicados en el artículo del workshop WASSA 2013 (Hernández et al., 2013) y se describen con más detalle en el Capítulo 2. Los resultados se presentan de manera resumida en la siguiente tabla.

Tabla 1.1: Resultados de la aplicación del método de PU-Learning.

Entrenamiento	precisión	cobertura	f-measure
80-Falsas/520-No etiquetadas	0.91	0.39	0.54
100-Falsas/520-No etiquetadas	0.90	0.43	0.58
120-Falsas/520-No etiquetadas	0.92	0.55	0.69

En la Tabla 1.1 se puede observar que se incrementa el valor del f-measure obtenido, conforme se aumenta el tamaño del conjunto de opiniones falsas. Estos valores se compararan ahora contra una variante del método de PU-Learning en el siguiente sub-apartado.

1.6.2. Variante del método PU-Learning

La variante del método de PU-Learning mejora en la manera como se obtiene el conjunto de opiniones verdaderas. El método tiene como característica principal que se van descartando las opiniones falsas del conjunto de opiniones no-etiquetadas, hasta quedarse con las opiniones verdaderas confiables. La ventaja de hacerlo de esta manera es que en cada iteración se van extrayendo las instancias que se etiquetan como opiniones falsas (es decir, que solo permanecen en el conjunto no etiquetado las opiniones verdaderas). Este proceso se lleva a cabo hasta que se cumple la condición de paro, la cual consiste en que ya no existan más opiniones falsas que se puedan

extraer del conjunto no-etiquetado.

En el trabajo publicado en el año 2013, "Using PU-Learning to detect deceptive Opinion Spam" (Hernández et al., 2013) que se describe con más detalle en el Capítulo 2, se realizó la evaluación sobre un conjunto de opiniones acerca de hoteles situados en la ciudad de Chicago. Se hicieron varios experimentos para evaluar la efectividad de la variante del método de PU-Learning cambiando el número de opiniones falsas del conjunto de entrenamiento y manteniendo fijo el número de elementos de la clase no-etiquetada, a partir de los cuales se empieza a iterar hasta que se consigue llegar a una condición de paro, donde ya no es posible extraer más instancias del conjunto no etiquetado.

Los resultados se presentan de manera resumida en la siguiente tabla.

Tabla 1.2: Resultados de la aplicación del método de PU-Learning*.

Entrenamiento	precisión	cobertura	f-measure
80-Falsas/520-No etiquetadas	0.87	0.41	0.56
100-Falsas/520-No etiquetadas	0.78	0.90	0.84
120-Falsas/520-No etiquetadas	0.79	0.78	0.78

Es posible observar a partir de la Tabla 1.2 que el mejor valor del f-measure promedio obtenido fue de 0.84. Este valor es muy comparable con los resultados obtenidos por otros métodos solo que con la diferencia que ahora se requirió un número menor de opiniones etiquetadas como falsas *(100 opiniones únicamente)*.

Estos buenos resultados dieron pie a seguir probando el nuevo método en otros corpus más grandes y diferentes. Con estos resultados se obtuvo la publicación "Detecting positive and negative opinions using PU-Learning" (Hernández et al., 2015a) que se presenta en el Capítulo 3.

Otra característica del nuevo método es la de funcionar con cualquier tipo de atributos. Se hicieron pruebas con otra clase de atributos y se presentan en el siguiente sub-apartado.

1.6.3. Método PU-Learning*

En la modificación realizada al método de PU-Learning, descrita en la sección anterior, se van extrayendo las opiniones falsas del conjunto no-etiquetado, con la finalidad de quedarnos únicamente con el conjunto de opiniones verdaderas confiables. Este proceso mejora los resultados obtenidos con respecto al método de PU-Learning, pero provoca un desbalanceo entre las clases (Japkowicz and Shaju, 2002), disminuyendo el desempeño del sistema de clasificación automático.

El método de PU-Learning* además de que presenta la mejora de cómo se obtiene el conjunto de opiniones verdaderas confiables, se le ha agregó una condición de paro que previene el desbalanceo que se presenta en algunos casos, tras haber realizado las iteraciones donde se extraen las opiniones clasificadas como falsas. De esta forma se logra obtener un clasificador que presenta un mejor desempeño para la detección de las opiniones falsas.

En el trabajo publicado en el año 2015, "Detection of opinion spam with character n-grams" (Hernández et al., 2015b) que se presenta en el Capítulo 4, se muestra el uso del método de PU-Learning* combinado con el uso de n-gramas de caracteres como atributos. En este trabajo se realizó la evaluación sobre el mismo conjunto de opiniones falsas favorables empleado en el experimento anterior, pero ademas se le agregaron las opiniones falsas desfavorables acerca de hoteles situados en el centro de la ciudad de Chicago. Se hicieron varios experimentos para evaluar la efectividad del método usando como atributos los n-gramas de caracteres en el rango de 3 a 5, que son los recomendados para el idioma inglés.

Como se puede observar en la tabla Tabla 1.3, la combinación del método de PU-Learning* y los atributos de n-gramas de caracteres obtiene mejores resultados de precisión, cuando se usan n-gramas de 5 caracteres para las opiniones favorables y n-gramas de 4 caracteres cuando se evalúan las opiniones desfavorables.

Tabla 1.3: Resultados de la aplicación del método de PU-Learning* con n-gramas de caracteres.

Polaridad	atributos	precisión	cobertura	f-measure
favorables	3-caracteres	0.840	0.853	0.846
	4-caracteres	0.895	0.870	0.882
	5-caracteres	0.897	0.910	0.903
desfavorables	3-caracteres	0.878	0.755	0.812
	4-caracteres	0.899	0.758	0.822
	5-caracteres	0.849	0.813	0.830

1.6.4. Detección de opiniones falsas en dominios cruzados

Uno de los mayores problemas que se presentan a la hora de llevar a cabo la detección de opiniones falsas, es el de contar con un conjunto de instancias etiquetadas del dominio que se quiere probar y que nos permitan llevar a cabo el aprendizaje del sistema de clasificación automática. Ante esta problemática nos planteamos el llevar a cabo pruebas de clasificación en dominios cruzados.

El método de PU-Learning* fue empleado para la detección de opiniones falsas favorables en ambientes de dominios cruzados. Este tipo de evaluación del método de PU-Learning* es apropiada cuando no se cuenta con opiniones etiquetadas de algún dominio en particular, pero sí de otro dominio diferente. Esto se presenta como un problema adicional en la detección de opiniones falsas, aumentando la complejidad del problema.

Los resultados obtenidos se presentan de una manera breve en las tablas 1.4 y 1.5, donde se puede observar que cuando existe cierta afinidad en el lenguaje empleado en los dominios involucrados, como es el caso de los dominios de hoteles y restaurantes, se obtienen mejores resultados que cuando existe poca afinidad.

En el Capítulo 5 se presenta el uso del método de PU-Learning* combinado con el uso de n-gramas de caracteres, para la detección de opiniones falsas en dominios cruzados con mayor detalle.

Tabla 1.4: Resultados de la aplicación del método de PU-Learning* en dominios cruzados con hoteles y restaurantes.

Entrenamiento hoteles		**Prueba restaurantes**		**Atributos**		**f-measure**
falsas	**verdaderas**	**falsas**	**verdaderas**	**tipo**	**cantidad**	**promedio**
400	400	90	200	3-CARACTERES	7963	0.791
				4-CARACTERES	26210	0.832
				5-CARACTERES	60797	0.774
100	140	90	200	3-CARACTERES	5215	0.681
				4-CARACTERES	15454	**0.749**
				5-CARACTERES	31426	0.717

Tabla 1.5: Resultados de la aplicación del método de PU-Learning* en dominios cruzados con hoteles y médicos.

Entrenamiento hoteles		**Prueba médicos**		**Atributos**		**f-measure**
falsas	**verdaderas**	**falsas**	**verdaderas**	**tipo**	**cantidad**	**promedio**
100	140	50	200	3-CARACTERES	7963	0.472
				4-CARACTERES	26210	0.479
				5-CARACTERES	60797	0.441
100	140	50	200	3-CARACTERES	5215	0.514
				4-CARACTERES	15454	0.546
				5-CARACTERES	31426	**0.605**

En el siguiente sub-apartado se presenta una descripción global del contenido de la tesis.

1.7. Organización de la tesis

Este trabajo de tesis está organizado en seis capítulos: en el Capítulo 1 se presenta la introducción al tema de la detección de opiniones falsas, los siguientes tres capítulos (Capítulo 2, Capítulo 3 y Capítulo 4), corresponden a las publicaciones realizadas sobre el nuevo método de detección de opiniones falsas basado en PU-Learning que se propone. En el Capítulo 5 se realiza la discusión de los resultados obtenidos y por último en el Capítulo 6 se presentan las conclusiones y trabajo futuro.

Una breve síntesis del contenido de los capítulos se muestra a continuación:

- Capítulo 2: se presenta el trabajo "Using PU-Learning to detect deceptive Opinion Spam" publicado en el workshop WASSA 2013. En este artículo se utiliza el método de PU-Learning* por primera vez, evaluándolo en un conjunto de opiniones favorables (falsas y verdaderas) sobre hoteles situados en el centro de la ciudad de Chicago. En esta publicación se compara el nuevo método con otros que requieren un conjunto completamente etiquetado de opiniones para su funcionamiento y se obtienen valores de f-measure promedio de 0.84.

- Capítulo 3: se presenta el artículo "Detecting positive and negative opinions using PU-Learning" publicado en la revista Information Processing & Management, donde se evalúa el método de PU-Learning* en un conjunto de opiniones que contiene tanto opiniones favorables como desfavorables de 20 hoteles del área del centro de Chicago. En este artículo se compara la precisión obtenida usando el método de PU-Learning* contra el método de PU-Learning, empleando diferentes tipos de atributos. También se realiza un análisis de significancia estadística entre los atributos empleados, para determinar la mejor combinación que nos lleve a obtener resultados con la precisión más alta.

- Capítulo 4: se presenta la publicación de la conferencia CICLing 2015 "Detection of opinion spam with character n-grams", donde se evalúa y compara

el método de PU-Learning* para detectar las opiniones falsas utilizando diferentes tipos de atributos, particularmente n-gramas de palabras y n-gramas de caracteres. En este trabajo se presenta también un análisis sobre la robustez de los n-gramas de caracteres en la clasificación con pocos datos de entrenamiento.

- Capítulo 5: se presenta una descripción completa de los corpus utilizados en los experimentos realizados para evaluar el método de PU-Learning*, así como los resultados obtenidos en los experimentos diseñados para probar la eficacia del método. También se realizan pruebas de dominios cruzados, donde se entrena en un dominio diferente al que se prueba.

- Capítulo 6: se presentan las conclusiones del desarrollo de este trabajo de tesis, el trabajo futuro y la lista de las publicaciones generadas.

Capítulo 2

Using PU-Learning to detect deceptive opinion spam

A continuación se presenta la versión de autor del artículo "Using PU-Learning to detect deceptive opinion spam", publicado en el 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis for Computational Linguistics, NAACL, páginas 38-45, 2013.

Abstract Nowadays a large number of opinion reviews are posted on the Web. Such reviews are a very important source of information for customers and companies. The former rely more than ever on online reviews to make their purchase decisions and the latter to respond promptly to their clients' expectations. Due to the economic importance of these reviews there is a growing trend to incorporate spam on such sites, and, as a consequence, to develop methods for opinion spam detection. In this paper we focus on the detection of *deceptive opinion spam*, which consists of fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers. In particular we propose a method based on the PU-learning approach which learns only from a few positive examples and a set of unlabeled data. Evaluation results in a corpus of hotel reviews demonstrate the appropriateness of the proposed method for real applications since it reached a f-measure of 0.84 in the detection of deceptive opinions using only 100 positive examples for training.

2.1. Introduction

The Web is the greatest repository of digital information and communication platform ever invented. People around the world widely use it to interact with each other as well as to express opinions and feelings on different issues and topics. With the increasing availability of online review sites and blogs, costumers rely more than ever on online reviews to make their purchase decisions and businesses to respond promptly to their clients' expectations. It is not surprising that opinion mining technologies have been witnessed a great interest in recent years (Mihalcea and Strapparava., 2009; Zhou et al., 2008). Research in this field has been mainly oriented to problems such as opinion extraction (Liu., 2012) and polarity classification (Reyes and Rosso., 2012). However, because of the current trend about the growing number of online reviews that are fake or paid by companies to promote their products or damage the reputation of competitors, the automatic detection of opinion spam has emerged as a highly relevant research topic (Jindal and Liu., 2008; Jindal et al., 2010; Ott et al., 2011; Raymond et al., 2011b; Sihong et al., 2012; Wu et al., 2010).

Detecting opinion spam is a very challenging problem since opinions expressed in the Web are typically short texts, written by unknown people using different styles

and for different purposes. Opinion spam has many forms, e.g., fake reviews, fake comments, fake blogs, fake social network postings and deceptive texts. Opinion spam reviews may be detected by methods that seek for duplicate reviews (Jindal and Liu., 2008), however, this kind of opinion spam only represents a small percentage of the opinions from review sites. In this paper we focus on a potentially more insidious type of opinion spam, namely, *deceptive opinion spam*, which consists of fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers.

The detection of deceptive opinion spam has been traditionally solved by means of supervised text classification techniques (Ott et al., 2011). These techniques have demonstrated to be very robust if they are trained using large sets of labeled instances from both classes, deceptive opinions (positive instances) and truthful opinions (negative examples). Nevertheless, in real application scenarios it is very difficult to construct such large training sets and, moreover, it is almost impossible to determine the authenticity of the opinions (Mukherjee et al., 2011). In order to meet this restriction we propose a method that learns only from a few positive examples and a set of unlabeled data. In particular, we propose applying the PU-Learning approach (Liu et al., 2002, 2003) to detect deceptive opinion spam.

The evaluation of the proposed method was carried out using a corpus of hotel reviews under different training conditions. The results are encouraging; they show the appropriateness of the proposed method for being used in real opinion spam detection applications. It reached a f-measure of 0.84 in the detection of deceptive opinions using only 100 positive examples, greatly outperforming the effectiveness of the traditional supervised approach and the one-class SVM model.

The rest of the paper is organized as follows. Section 2.2 presents some related works in the field of opinion spam detection. Section 2.3 describes our adaptation of the PU-Learning approach to the task of opinion spam detection. Section 2.4 presents the experimental results and discusses its advantages and disadvantages. Finally, Section 2.5 indicates the contributions of the paper and provides some future work directions.

2.2. Related Work

The detection of spam in the Web has been mainly approached as a binary classification problem (spam vs. non-spam). It has been traditionally studied in the context of e-mail (Drucker et al., 2002), and web pages (Gyongyi et al., 2004; Ntoulas et al., 2006). The detection of opinion spam, i.e., the identification of fake reviews that try to deliberately mislead human readers, is just another face of the same problem (Raymond et al., 2011b). Nevertheless, the construction of automatic detection methods for this task is more complex than for the others since manually gathering labeled reviews –particularly truthful opinions– is very hard, if not impossible (Mukherjee et al., 2011).

One of the first works regarding the detection of opinion spam reviews was proposed by (Jindal and Liu., 2008). He proposed detecting opinion spam by identifying duplicate content. Although this method showed good precision in a review data set from Amazon[1], it has the disadvantage of under detecting original fake reviews. It is well known that spammers modify or paraphrase their own reviews to avoid being detected by automatic tools.

In (Wu et al., 2010), the authors present a method to detect hotels which are more likely to be involved in spamming. They proposed a number of criteria that might be indicative of suspicious reviews and evaluated alternative methods for integrating these criteria to produce a suspiciousness ranking. Their criteria mainly derive from characteristics of the network of reviewers and also from the impact and ratings of reviews. It is worth mentioning that they did not take advantage of reviews' content for their analysis.

Ott et al. (Ott et al., 2011) constructed a classifier to distinguish between deceptive and truthful reviews. In order to train their classifier they considered certain types of near duplicates reviews as positive (deceptive) training data and *the rest* as the negative (truthful) training data. The review spam detection was done using different stylistic, syntactical and lexical features as well as using SVM as base classifier.

In a recent work, Sihong et al. (Sihong et al., 2012) demonstrated that a high

[1]http://www.Amazon.com

correlation between the increase in the volume of (singleton) reviews and a sharp increase or decrease in the ratings is a clear signal that the rating is manipulated by possible spam reviews. Supported by this observation they proposed a spam detection method based on time series pattern discovery.

The method proposed in this paper is similar to Ott's et al. method in the sense that it also aims to automatically identify deceptive and truthful reviews. However, theirs shows a key problem: it depends on the availability of labeled negative instances which are difficult to obtain, and that causes traditional text classification techniques to be ineffective for real application scenarios. In contrast, our method is specially suited for this application since it builds accurate two-class classifiers with only positive and unlabeled examples, but not negative examples. In particular we propose using the PU-Learning approach (Liu et al., 2002, 2003) for opinion spam detection. To the best of our knowledge this is the first time that this technique, or any one-class classification approach, has been applied to this task. In (Ferretti et al., 2012) PU-learning was successfully used in the task of Wikipedia flaw detection[2].

2.3. PU-Learning for opinion spam detection

PU-learning is a partially supervised classification technique. It is described as a two-step strategy which addresses the problem of building a two-class classifier with only positive and unlabeled examples (Liu et al., 2002, 2003; Zhang and Zuo, 2009). Broadly speaking this strategy consists of two main steps: *i*) to identify a set of reliable negative instances from the unlabeled set, and *ii*) to apply a learning algorithm on the refined training set to build a two-class classifier.

Figure 2.1 shows our adaptation of the PU-learning approach for the task of opinion spam detection. The proposed method is an iterative process with two steps. In the first step the whole unlabeled set is considered as the negative class. Then, we train a classifier using this set in conjunction with the set of positive examples. In the second step, this classifier is used to classify (automatically label) the unlabeled set. The instances from the unlabeled set classified as positive are eliminated; the rest of them are considered as the reliable negative instances for the next iteration.

[2]http://www.webis.de/research/events/pan-12

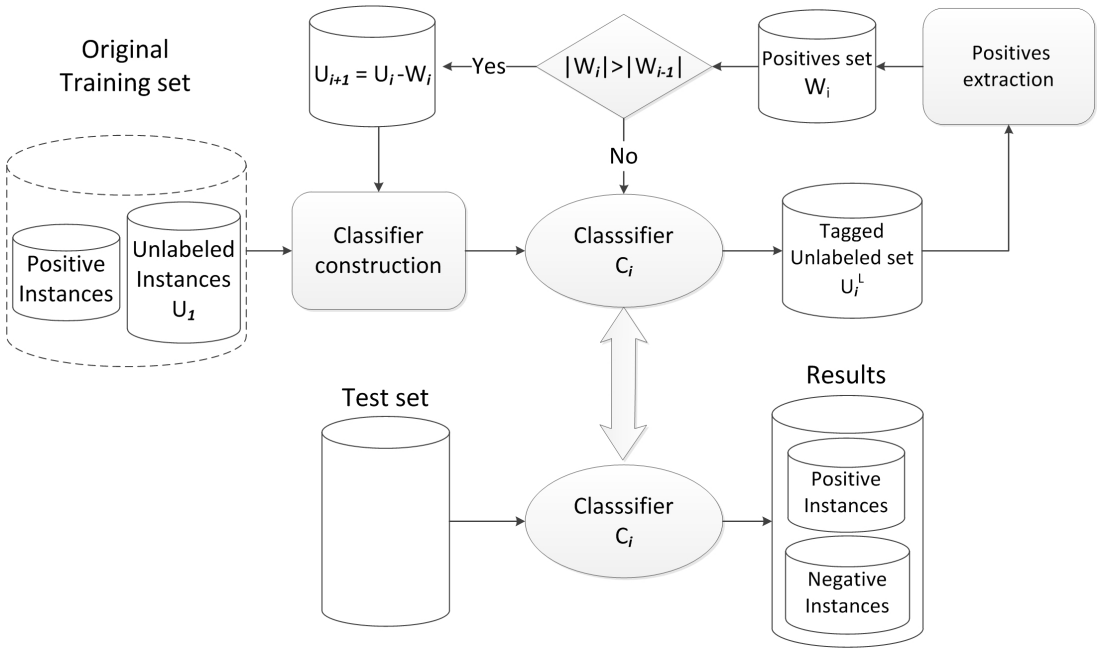

Figure 2.1: Classifier construction with PU-Learning approach.

This iterative process is repeated until a stop criterion is reached. Finally, the latest built classifier is returned as the final classifier.

In order to clarify the construction of the opinion spam classifier, Algorithm 3 presents the formal description of the proposed method. In this algorithm P is the set of positive instances and U_i represents the unlabeled set at iteration i; U_1 is the original unlabeled set. C_i is used to represent the classifier that was built at iteration i, and W_i indicates the set of unlabeled instances classified as positive by the classifier C_i. These instances have to be removed from the training set for the next iteration. Therefore, the negative class for next iteration is defined as $U_i - W_i$. Line 4 of the algorithm shows the stop criterion that we used in our experiments, $|W_i| <= |W_{i-1}|$. The idea of this criterion is to allow a continue but gradual reduction of the negative instances.

Algorithm 3 PU-Learning for opinion spam detection

1: $i \leftarrow 1$
2: $|W_0| \leftarrow |U_1|$
3: $|W_1| \leftarrow |U_1|$
4: **while** $|W_i| <= |W_{i-1}|$ **do**
5: $\quad C_i \leftarrow Generate\_Classifier(P, U_i)$
6: $\quad U_i^L \leftarrow C_i(U_i)$
7: $\quad W_i \leftarrow Extract\_Positives(U_i^L)$
8: $\quad U_{i+1} \leftarrow U_i - W_i$
9: $\quad i \leftarrow i + 1$
10: Return Classifier C_i

2.4. Evaluation

2.4.1. Datasets

The evaluation of the proposed method was carried out using a dataset of reviews assembled by Ott et al. (Ott et al., 2011). This corpus contains 800 opinions, 400 deceptive and 400 truthful opinions. These opinions are about the 20 most popular Chicago hotels; deceptive opinions were generated using the Amazon Mechanical Turk (AMT)[3], whereas –possible– truthful opinions were mined from a total of 6,977 reviews on TripAdvisor[4]. The following paragraphs show two opinions taken from (Ott et al., 2011). These examples are very interesting since they show the great complexity of the automatically –and even manually– detection of deceptive opinions. Both opinions are very similar and just minor details can help distinguishing one from the other. For example, in his research Ott et al. (Ott et al., 2011) found that deceptive reviews used the words "experience", "my husband", "I", "feel", "business", and "vacation"more than genuine ones.

Example of a truthful opinion

We stay at Hilton for 4 nights last march. It was a pleasant stay. We got a large room with 2 double beds and 2 bathrooms, The TV was Ok, a 27' CRT Flat Screen. The concierge was very

[3]http://www.mturk.com
[4]http://www.tripadvisor.com

friendly when we need. The room was very cleaned when we arrived, we ordered some pizzas from room service and the pizza was ok also. The main Hall is beautiful. The breakfast is charged, 20 dollars, kinda expensive. The internet access (WiFi) is charged, 13 dollars/day. Pros: Low rate price, huge rooms, close to attractions at Loop, close to metro station. Cons: Expensive breakfast, Internet access charged. Tip: When leaving the building, always use the Michigan Ave exit. It's a great view.

Example of a deceptive opinion

My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free WiFi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.

In order to simulated real scenarios to test our method we assembled several different sub-corpora from Ott's et al. (Ott et al., 2011) dataset. First we randomly selected 80 deceptive opinions and 80 truthful opinions to build a fixed test set. The remaining 640 opinions were used to build six training sets of different sizes and distributions. They contain 20, 40, 60, 80, 100 and 120 positive instances (deceptive opinions) respectively. In all cases we used a set of 520 unlabeled instances containing a distribution of 320 truthful opinions and 200 deceptive opinions.

2.4.2. Evaluation Measure

The evaluation of the effectiveness of the proposed method was carried out by means of the f-measure. This measure is a linear combination of the precision and recall values. We computed this measure for both classes, deceptive and –possible– truthful opinions, nevertheless, the performance on the deceptive opinions is the only measure of real relevance. The f-measure for each opinion category O_i is defined as follows:

$$f - measure(O_i) = \frac{2 \times recall(O_i) \times precision(O_i)}{recall(O_i) + precision(O_i)} \tag{2.1}$$

$$recall(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ opinions\ of\ O_i} \tag{2.2}$$

$$precision(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ predictions\ as\ O_i} \tag{2.3}$$

2.4.3. Results

Tables 2.1 and 2.2 show the results from all the experiments we carried out. It is important to notice that we used Naïve Bayes and SVM classifiers as learning algorithms in our PU-learning method. These learning algorithms as well as the one-class implementation of SVM were also used to generated baseline results. In all the experiments we used the default implementations of these algorithms in the Weka experimental platform (Hall et al., 2009).

In order to make easy the analysis and discussion of the results we divided them in three groups: baseline results, one-class classification results, and PU-learning results. The following paragraphs describe these results.

Baseline results: The baseline results were obtained by training the NB and SVM classifiers using the unlabeled dataset as the negative class. This is a common approach to build binary classifiers in lack of negative instances. It also corresponds to the results of the first iteration of the proposed PU-learning based method. The rows named as "BASE NB" and "BASE SVM" show these results. They results clearly indicate the complexity of the task and the inadequacy of the traditional classification approach. The best f-measure in the deceptive opinion class (0.68) was obtained by the NB classifier when using 120 positive opinions for training. For the cases considering less number of training instances this approach generated very poor results. In addition we can also noticed that NB outperformed SVM in all cases.

One-class classification results: These results correspond to the application of the one-class SVM learning algorithm (Manevitz and Yousef., 2002), which is a very robust approach for this kind of problems. This algorithm only uses the positive examples to build the classifier and does not take advantage of the available unlabeled instances. Its results are shown in the rows named as "ONE CLASS"; these results are very interesting since clearly show that this approach is very robust when there are only some examples of deceptive opinions (please refer to Table 2.1). On the

contrary, it is also clear that this approach was outperformed by others, especially by our PU-learning based method, when more training data was available.

PU-Learning results: Rows labeled as "PU-LEA NB.ªnd "PU-LEA SVM"show the results of the proposed method when the NB and SVM classifiers were used as base classifiers respectively. These results indicate that: *i*) the application of PU-learning improved baseline results in most of the cases, except when using 20 and 40 positive training instances; *ii*) PU-Learning results clearly outperformed the results from the one-class classifier when there were used more than 60 deceptive opinions for training; *iii*) results from "PU-LEA NB"were usually better than results from "PU-LEA SVM". It is also important to notice that both methods quickly converged, requiring less than seven iterations for all cases. In particular, "PU-LEA NB"took more iterations than "PU-LEA SVM", leading to greater reductions of the unlabeled sets, and, consequently, to a better identification of the subsets of reliable negative instances.

Original Training Set	**Approach**	**Truthful**			**Deceptive**			**Iteration**	**Final Training Set**
		P	**R**	**F**	**P**	**R**	**F**		
20-D 520-U	One class	0.500	0.688	0.579	*0.500*	*0.313*	*0.385*		
	base NB	0.506	1.000	0.672	1.000	0.025	0.049		
	PU-LEA NB	0.506	1.000	0.672	1.000	0.025	0.049	5	20-D/493- U
	base SVM	0.500	1.000	0.667	0.000	0.000	0.000		
	PU-LEA SVM	0.500	1.000	0.667	0.000	0.000	0.000	4	20-D/518-U
40-D 520-U	One class	0.520	0.650	0.578	*0.533*	*0.400*	*0.457*		
	base NB	0.517	0.975	0.675	0.778	0.088	0.157		
	PU-LEA NB	0.517	0.975	0.675	0.778	0.088	0.157	4	40-D/479-U
	base SVM	0.519	1.000	0.684	1.000	0.075	0.140		
	PU-LEA SVM	0.516	0.988	0.678	0.857	0.075	0.138	3	40-D/483-U
60-D 520-U	One class	0.500	0.500	0.500	*0.500*	*0.500*	*0.500*		
	base NB	0.569	0.975	0.719	0.913	0.263	0.408		
	PU-LEA NB	0.574	0.975	0.722	0.917	0.275	0.423	3	60-D/449-U
	base SVM	0.510	0.938	0.661	0.615	0.100	0.172		
	PU-LEA SVM	0.517	0.950	0.670	0.692	0.113	0.194	3	60-D/450-U

Table 2.1: Comparison of the performance of different classifiers when using 20, 40 and 60 examples of deceptive opinions for training; in this table D refers to deceptive opinions and U to unlabeled opinions.

Finally, Figure 2.2 presents a summary of the best results obtained by each of the methods in all datasets. From this figure it is clear the advantage of the one-class

Original Training Set	Approach	Truthful			Deceptive			Iteration	Final Training Set
		P	R	F	P	R	F		
80-D	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.611	0.963	0.748	0.912	0.388	0.544		
520-D	PU-LEA NB	0.615	0.938	0.743	*0.868*	*0.413*	*0.559*	6	80-D/267-U
	BASE SVM	0.543	0.938	0.688	0.773	0.213	0.333		
	PU-LEA SVM	0.561	0.925	0.698	0.786	0.275	0.407	3	80-D/426-U
100-D	ONE CLASS	0.482	0.513	0.497	0.480	0.450	0.465		
	BASE NB	0.623	0.950	0.752	0.895	0.425	0.576		
520-U	PU-LEA NB	0.882	0.750	0.811	**0.783**	**0.900**	**0.837**	7	100-D/140-U
	BASE SVM	0.540	0.938	0.685	0.762	0.200	0.317		
	PU-LEA SVM	0.608	0.913	0.730	0.825	0.413	0.550	4	100-D/325-U
120-D	ONE CLASS	0.494	0.525	0.509	0.493	0.463	0.478		
	BASE NB	0.679	0.950	0.792	0.917	0.550	0.687		
520-U	PU-LEA NB	0.708	0.850	0.773	*0.789*	*0.781*	*0.780*	5	120-D/203-U
	BASE SVM	0.581	0.938	0.718	0.839	0.325	0.468		
	PU-LEA SVM	0.615	0.738	0.670	0.672	0.538	0.597	6	120-D/169-U

Table 2.2: Comparison of the performance of different classifiers when using 80, 100 and 120 examples of deceptive opinions for training; in this table D refers to deceptive opinions and U to unlabeled opinions.

SVM classifier when having only some examples of deceptive opinions for training, but also it is evident the advantage of the proposed method over the rest when having a considerable quantity of deceptive opinions for training. It is important to emphasize that the best result obtained by the proposed method (a F-meausre of 0.837 in the deceptive opinion class) is a very important results since it is comparable to the best result (0.89) reported for this collection/task, but when using 400 positive and 400 negative instances for training. Moreover, this result is also far better than the best human result obtained in this dataset, which, according to (Ott et al., 2011) it is around 60 % of accuracy.

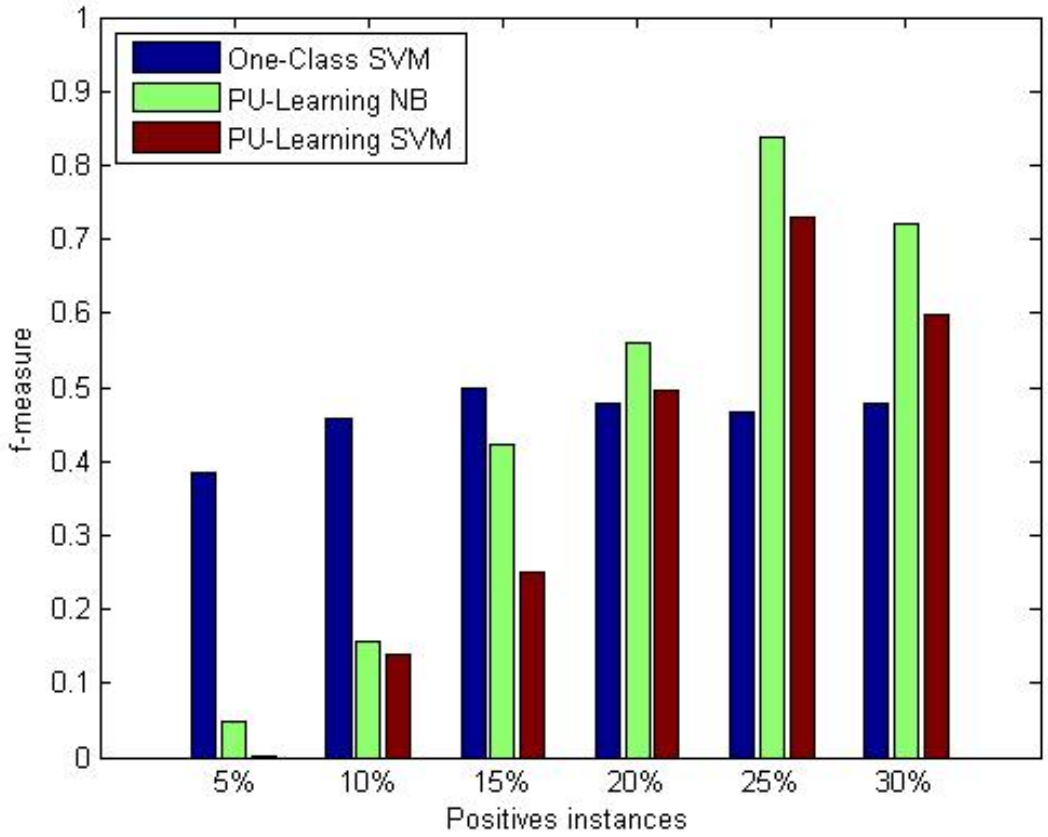

Figure 2.2: Summary of best Results; f-measure.

2.5. Conclusions and future work

In this paper we proposed a novel method for detecting deceptive opinion spam. This method adapts the PU-learning approach to this task. In contrast to traditional approaches that require large sets of labeled instances from both classes, deceptive and truthful opinions, to build accurate classifiers, the proposed method only uses a small set of deceptive opinion examples and a set of unlabeled opinions. This characteristic represents a great advantage of our method over previous approaches since in real application scenarios it is very difficult to construct such large training sets and, moreover, it is almost impossible to determine the authenticity or truthfulness

of the opinions.

The evaluation of the method in a set of hotel reviews indicated that the proposed method is very appropriate for the task of opinion spam detection. It achieved a F-meausre of 0.837 in the classification of deceptive opinions using only 100 positive examples and a bunch of unlabeled instances for training. This result is very relevant since it is comparable to previous results obtained by highly supervised methods in similar evaluation conditions.

Another important contribution of this work was the evaluation of a one-class classifier in this task. For the experimental results we can conclude that the usage of a one-class SVM classifier is very adequate for cases when there are only very few examples of deceptive opinions for training. In addition we could observe that this approach and the proposed method based on PU-learning are complementary. The one-class SVM classifier obtained the best results using less than 50 positive training examples, whereas the proposed method achieved the best results for the cases having more training examples.

As future work we plan to integrate the PU-learning and self-training approaches. Our idea is that iteratively adding some of the unlabeled instances into the original positive set may further improve the classification accuracy. We also plan to define and evaluate different stop criteria, and to apply this method in other related tasks such as email spam detection or phishing url detection.

Acknowledgments

This work is the result of the collaboration in the framework of the WIQEI IRSES project (Grant No. 269180) within the FP 7 Marie Curie. The work of the last author was in the framework the DIANA-APPLICATIONS-Finding Hidden Knowledge in Texts: Applications (TIN2012-38603-C02-01) project, and the VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

Capítulo 3

Detecting positive and negative deceptive opinions using PU-Learning

A continuación se presenta la versión de autor del artículo "Detecting positive and negative deceptive opinions using PU-Learning", publicado en la revista Information Processing & Management, Volumen 51, Issue 4, páginas 433-443, http://dx.doi.org/10.1016/j.ipm.2014.11.001, 2015.

En este artículo se presentan los resultados obtenidos por el método propuesto de PU-Learning cuando se emplean como atributos los n-gramas de palabras.*

Abstract

Nowadays a large number of opinion reviews are posted on the Web. Such reviews are a very important source of information for customers and companies. The former rely more than ever on online reviews to make their purchase decisions, and the latter to respond promptly to their clients' expectations. Unfortunately, due to the business that is behind, there is an increasing number of deceptive opinions, that is, fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers promoting a low quality product (positive deceptive opinions) or criticizing a potentially good quality one (negative deceptive opinions). In this paper we focus on the detection of both types of deceptive opinions, *positive and negative*. Due to the scarcity of examples of deceptive opinions, we propose to approach the problem of the detection of deceptive opinions employing PU-learning. PU-learning is a semi-supervised technique for building a binary classifier on the basis of positive (i.e., deceptive opinions) and unlabeled examples only. Concretely, we propose a *novel* method that with respect to its original version is much more conservative at the moment of selecting the negative examples (i.e., *not* deceptive opinions) from the unlabeled ones. The obtained results show that the proposed PU-learning method *consistently* outperformed the original PU-learning approach. In particular, results show an average improvement of 8.2 % and 1.6 % over the original approach in the detection of positive and negative deceptive opinions respectively.

Keywords Opinion Mining, Opinion Spam, Deceptive Opinions, PU-learning.

3.1. Introduction

The Web is not only the greatest repository of digital information ever invented but also the largest communication platform. This characteristic has motivated businesses of all sizes and kinds, such as television networks, film makers, hotels and restaurants, to use the Web as a critical marketing venue by creating websites and discussion forums for their products and services (Duan et al., 2008). With the increasing availability of such review sites and blogs, consumers rely more than ever on online reviews to make their purchase decisions. A recent survey found that 87 %

of them have reinforced their decisions to purchase a product or service by positive online reviews. At the same time, 80 % of consumers have also changed their minds about purchases based on negative information they found online[1].

Detecting opinion spam is a very challenging problem since opinions expressed on the Web are typically short texts, written by unknown people using different styles and for different purposes. Opinion spam has many forms, e.g., fake reviews, fake comments, fake blogs, fake social network postings and deceptive texts. Opinion spam reviews may be detected by methods that seek for duplicate reviews (Jindal and Liu., 2008); however, this kind of opinion spam only represents a small percentage of the opinions from review sites. In this paper we focus on a potentially more insidious type of opinion spam, namely, *deceptive opinion spam*, which consists of fictitious opinions that have been deliberately written to sound authentic in order to deceive the consumers.

The detection of deceptive opinion spam has been recently solved by means of supervised text classification techniques. These techniques have demonstrated to be very robust if they are trained using large sets of labeled instances from both classes, deceptive and truthful opinions. For example, some works have reported F_1 measures around 0.90 (Feng and Hirst, 2013; Ott et al., 2011, 2013). Nevertheless, in real application scenarios it is very difficult to construct such large training sets and, much more important, it is almost impossible to determine the authenticity of the opinions, i.e., to assemble a set of verified truthful reviews (Mukherjee et al., 2011). In order to meet this restriction in this paper we propose to apply *PU-learning* (Liu et al., 2002) to detect deceptive opinion spam in order to be able to learn only from a few examples of deceptive opinions and a set of unlabeled data, under the consideration that deceptive opinion spam can be accurately generated using a Mechanical Turk crowdsourcing service as suggested by (Ott et al., 2011).

The PU-learning approach was originally used and evaluated in thematic text classification, in problems showing high cohesion among the documents from the target (positive) class, and having great diversity in the unlabeled subset (Liu et al., 2002, 2003). The main contribution of this paper is the proposal of a conservative variant of the original method by (Liu et al., 2002) that is especially suited to the task

[1]How Online Reviews Affect Your Business. http://mwpartners.com/positive-online-reviews. Visited: April 2, 2014.

of detection of opinion spam, where deceptive opinions are very diverse in content and style, and there are only slightly differences between deceptive and truthful opinions.

The evaluation of the proposed method was carried out using a set of hotel reviews gathered by (Ott et al., 2013) containing *positive* and *negative* deceptive opinion spam[2]. The results are encouraging; on the one hand, they indicate that using only a hundred of examples of deceptive opinions for training it is possible to reach classification F_1 measures of 0.8 and 0.7 for positive and negative opinions respectively. On the other hand, they demonstrate the appropriateness of the proposed PU-learning variant for detecting opinion spam, since its results significantly outperformed those from the original approach in both kinds of opinion spam. As a further contribution, in a last experiment we analysed the role of opinions' polarity in the detection of deception. Our results confirm that negative deceptive opinions are more difficult to detect than positive spam, but they also show that having one single classifier for analysing both kinds of opinions is better that using two separate classifiers, suggesting that there are common characteristics in the way people write positive and negative opinion spam.

The rest of the paper is organized as follows. Section 3.2 introduces some related works in the field of opinion spam detection. Section 3.3 describes our adaptation of the PU-learning approach to the task of opinion spam detection. Section 3.4 presents the different opinion spam datasets used in the experiments. Section 3.5 describes the experimental settings and presents the results from the classification of deceptive and truthful reviews in several sets of positive and negative opinions. Finally, Section 3.6 presents our conclusions and discusses some future work directions.

3.2. Related Work

The detection of spam on the Web has been mainly approached as a binary classification problem (spam vs. non-spam). It has been traditionally studied in the context of e-mail (Drucker et al., 2002), and Web pages (Gyongyi et al., 2004; Ntoulas et al., 2006). The detection of opinion spam, i.e., the identification of fake reviews

[2]http://myleott.com/op_spam

that try to deliberately mislead human readers, is just another face of the same problem (Raymond et al., 2011a). Nevertheless, the construction of automatic detection methods for this task is more complex than for the others since manually gathering labeled reviews –particularly truthful opinions– is very hard, if not impossible (Mukherjee et al., 2011).

Due to the lack of reliable labeled data, most initial works regarding the detection of opinion spam considered unsupervised approaches which relied on meta-information from reviews and reviewers. For example, (Jindal and Liu., 2008) proposed detecting opinion spam by identifying duplicate content. Although this method showed good precision in a review data set from Amazon, it has the disadvantage of under detecting original fake reviews. It is well known that spammers modify or paraphrase their own reviews to avoid being detected by automatic tools. In a subsequent paper, (Jindal et al., 2010) proposed to detect spammers by searching for unusual review patterns; for example, they classify a reviewer as spam suspect if he wrote negative reviews about all the products of a brand but wrote positive reviews about a competing brand.

In this same category of unsupervised approaches, (Mukherjee et al., 2011) proposed a method for detecting groups of opinion spammers based on criteria such as the number of products for which the group work together and a high content similarity of their reviews. Similarly, in (Wu et al., 2010) the authors present a method to detect hotels which are more likely to be involved in spamming. They proposed a number of criteria that might be indicative of suspicious reviews and evaluated alternative methods for integrating these criteria to produce a suspiciousness ranking. Their criteria mainly derive from characteristics of the network of reviewers and also from the impact and ratings of reviews. It is worth mentioning that they did not take advantage of reviews' content for their analysis. Finally, in a recent work by (Sihong et al., 2012), it has been demonstrated that a high correlation between the increase in the volume of singleton reviews and a sharp increase or decrease in the ratings is a clear signal that the rating is manipulated by possible spam reviews. Supported by this observation they proposed a spam detection method based on temporal pattern discovery.

It was only after the release of the gold-standard datasets by (Ott et al., 2011,

2013), which contain examples of positive and negative deceptive opinion spam, that it was possible to conduct supervised learning and a reliable evaluation of the task. (Ott et al., 2011) constructed a SVM classifier to distinguish between *positive* deceptive and truthful reviews using different stylistic, syntactic and lexical features. Then, in (Ott et al., 2013) they applied the same approach to classify *negative* opinions. The main conclusion from these works is that standard text categorization techniques using unigrams and bigrams word features are effective at detecting deception in text, and that their results significantly outperform those from human judges. Following this research direction, (Feng et al., 2012a,b) extended Ott et al.'s n-gram feature set by incorporating deep syntax features, i.e., syntactic production rules derived from Probabilistic Context Free Grammar (PCFG) parse trees. Their experimental results consistently find statistical evidence that deep syntactic patterns are helpful in discriminating deceptive writing. Similarly, (Feng and Hirst, 2013) extended Ott et al. and Feng et al.'s works by incorporating features that characterize the degree of compatibility between the personal experience described in a test review and a product profile derived from a collection of reference reviews about the same product. This idea was supported on the hypothesis that since the writer of a deceptive review usually does not have any actual experience with that product, the resulting review might contain some contradictions with facts about the product. This approach showed to significantly improve the performance of identifying deceptive reviews.

The method proposed in this paper is similar to the above-mentioned works in the sense that it also applies a supervised approach to automatically identify deceptive and truthful reviews. However, all these methods exhibit a key problem: they depend on the availability of large amounts of labeled examples of deceptive and truthful opinions. This is particularly evident for the last two works which look for syntactic patterns and profile features. In order to overcome this limitation and be able to deal with real application scenarios, in (Hernández et al., 2013) we proposed a method that learns only from a few examples of deceptive opinions and a set of unlabeled data. Specifically, we have evaluated the feasibility of detecting positive deceptive opinions with PU-learning. This paper extend our previous work in four ways: it compares the performance of the proposed approach and the original PU-

learning method in the classification of deceptive opinion spam; it reports additional experimental results on a set of negative deceptive opinions, showing the proficiency of the method to deal with opinion spam of both polarities; it studies the role of opinions' polarity in the detection of deception; lastly, it presents an analysis of the performance of the method when using word unigrams and bigrams as features as well as different classifiers, particularly SVM and Naïve Bayes.

3.3. PU-Learning for Opinion Spam Detection

PU-learning is a semi-supervised technique for building a binary classifier based on positive and unlabeled examples only (Liu et al., 2002, 2003). In PU-learning, two sets of examples are available for training: the set P of positive instances, and a set U, which is assumed to contain a mixture of both positive and negative examples, but without any label. This contrasts with other forms of semi-supervised learning, where it is assumed that the training set contains labeled examples of both classes. In our particular problem, P corresponds to the set of labeled deceptive opinions, and U is a set of unlabeled review opinions –presumably– containing a combination of deceptive and truthful opinions.

The basic algorithm for PU-learning as described in (Liu et al., 2002, 2003) is shown in Algorithm 4. From now on we will refer to this algorithm as *original* PU-learning. The first part of this algorithm (from line 1 to 6) considers the identification of an initial set of reliable negative instances from U. It proceeds as follows: first, the whole unlabeled set U is considered as the negative class, and a classifier is trained using this set in conjunction with the set P of positive examples. Then, this classifier is used to classify (i.e., automatically label) the unlabeled set U. The instances from the unlabeled set classified as negative are selected to form the initial set of reliable negative instances (RN). The second part of the algorithm (from line 7 to 13) iteratively enlarges the set of reliable negative instances by aggregating some additional instances from U. This is done by training a binary classifier using the sets P and RN (from the previous iteration), and classifying the remaining instances at U. The instances from U classified as negative (Q) are aggregated to the set of reliable negative instances from the previous iteration.

Algorithm 4 Original PU-learning algorithm. P and U are the sets of positive and unlabeled examples respectively; C_i is the binary classifier at iteration i; Q_i represents the set of unlabeled examples from U_i classified as negative by C_i, and RN_i is the set of reliable negative examples gathered from iteration 1 to iteration i.

1: $i \leftarrow 1$
2: $C_i \leftarrow Generate\_Classifier(P, U)$
3: $U_i^L \leftarrow C_i(U)$
4: $Q_i \leftarrow Extract\_Negatives(U_i^L)$
5: $RN_i \leftarrow Q_i$
6: $U_i \leftarrow U - Q_i$
7: **while** $|Q_i| > \emptyset$ **do**
8: $\quad i \leftarrow i + 1$
9: $\quad C_i \leftarrow Generate\_Classifier(P, RN_{i-1})$
10: $\quad U_i^L \leftarrow C_i(U_{i-1})$
11: $\quad Q_i \leftarrow Extract\_Negatives(U_i^L)$
12: $\quad U_i \leftarrow U_{i-1} - Q_i$
13: $\quad RN_i \leftarrow RN_{i-1} + Q_i$
14: $Return(C_i)$

The original PU-learning approach has shown very good performance in text classification (Liu et al., 2002, 2003). It has been observed that its effectiveness is very related to the level of cohesion among the positive examples. Accordingly, in tasks showing high similarity among the positive labeled examples, the PU-learning algorithm tends to do a good initial selection of the reliable negative instances and, iteration by iteration, it is able to enlarge this set with more relevant negative examples.

Motivated by this observation, and by the fact that deceptive opinions are very diverse in content and style, we propose a conservative variant of the original PU-learning algorithm. This new algorithm, herein referred as *modified* PU-learning, assumes that the first classifier will be somewhat imprecise and it may select a potentially very noisy initial set of reliable negative instances. Therefore, instead of following an iterative growing strategy for building the RN set, this method considers its iterative pruning. Algorithm 5 describes the modified PU-learning algorithm. The first part of this algorithm (from line 1 to 6) is the same as in the original algorithm. The second part of the algorithm (from line 7 to 12) is significantly different: it iteratively reduces the set of reliable negative instances by eliminating the less

confident instances from RN. This is done by training a binary classifier using the sets P and RN (from previous iteration), and classifying the instances at RN. The instances classified as positive are eliminated from it, forming in this way a new small set of reliable negative instances. Line 7 from the algorithm indicates the new stop condition. The purpose of this condition is two-fold: on the one hand, to ensure a continuous but gradual reduction of the instances from the unlabeled set used as negative examples, and, on the other hand, to avoid a high imbalance in the training set by a radical reduction of RN. By means of this condition it is possible to identify a few number of high quality negative instances from the unlabeled set, and to construct a better final binary classifier than using the original PU-learning approach.

Algorithm 5 Modified PU-learning algorithm. P and U are the sets of positive and unlabeled examples respectively; Q_i and RN_i represent the sets of identified and retained reliable negative examples at iteration i, and C_i is the binary classifier at iteration i.

1: $i \leftarrow 1$;
2: $C_i \leftarrow Generate\_Classifier(P, U)$
3: $U_i^L \leftarrow C_i(U)$
4: $Q_i \leftarrow Extract\_Negatives(U_i^L)$
5: $RN_i \leftarrow Q_i$
6: $Q_0 \leftarrow Q_i$
7: **while** ($|Q_i| <= |Q_{i-1}|$ and $|P| < |RN_i|$) **do**
8: $\quad i \leftarrow i + 1$
9: $\quad C_i \leftarrow Generate\_Classifier(P, RN_{i-1})$
10: $\quad RN_i^L \leftarrow C_i(RN_{i-1})$
11: $\quad Q_i \leftarrow Extract\_Negatives(RN_i^L)$
12: $\quad RN_i \leftarrow Q_i$
13: $Return(C_i)$

3.4. Datasets

The evaluation of the proposed method was carried out using the corpora assembled by (Ott et al., 2011, 2013). These corpora include a total of 1600 labeled examples of deceptive and truthful review opinions about the 20 most popular Chica-

go hotels[3]. The corpora is organized as follows: 400 truthful positive reviews, 400 truthful negative reviews, 400 deceptive positive reviews and 400 deceptive negative reviews. Deceptive opinions were generated using the Amazon Mechanical Turk, whereas (likely) truthful opinions were mined from reviews on TripAdvisor, Expedia, Hotels.com, Orbitz, Priceline, and Yelp. The following paragraphs show two positive opinions taken from (Ott et al., 2011). These examples are very interesting since they show the great complexity of the automatically –and even manually– detection of deceptive opinions. Both opinions are very similar and just minor details can help distinguishing one from the other. For example, in their research (Ott et al., 2011) found that there is a relationship between deceptive language and imaginative writing, and that deceptive reviews tend to use the words "experience", "my husband", "I", "feel", "business", and "vacation"more than genuine ones.

Example of a positive *deceptive* opinion

> My husband and I stayed for two nights at the Hilton Chicago, and enjoyed every minute of it! The bedrooms are immaculate, and the linens are very soft. We also appreciated the free WiFi, as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious, and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone.

Example of a positive *truthful* opinion

> We stay at Hilton for 4 nights last march. It was a pleasant stay. We got a large room with 2 double beds and 2 bathrooms, The TV was Ok, a 27' CRT Flat Screen. The concierge was very friendly when we need. The room was very cleaned when we arrived, we ordered some pizzas from room service and the pizza was ok also. The main Hall is beautiful. The breakfast is charged, 20 dollars, kinda expensive. The internet access (WiFi) is charged, 13 dollars/day. Pros: Low rate price, huge rooms, close to attractions at Loop, close to metro station. Cons: Expensive breakfast, Internet access charged. Tip: When leaving the building, always use the Michigan Ave exit. It's a great view.

In order to simulate real scenarios to evaluate the performance of the proposed PU-learning method we assembled several different datasets from Ott et al.'s corpora.

[3]http://myleott.com/op_spam

These datasets contain opinions from both polarities and different number of labeled samples for training. The following paragraphs describe their construction. It is worth mentioning that for the experiments we built *five* different examples for each subset configuration, and that we always report their average results.

Datasets of positive opinions: From the set of 400 deceptive and 400 truthful positive opinions from Ott et al.'s corpora, we first randomly selected 80 deceptive opinions and 80 truthful opinions to build a fixed test set. Then, the remaining 640 opinions were used to build six training sets of different sizes and distributions. They contain 20, 40, 60, 80, 100 and 120 positive instances (deceptive opinions) respectively. In all cases we used a set of 520 unlabeled instances containing a distribution of 320 truthful opinions and 200 positive deceptive opinions.

Datasets of negative opinions: Their construction was similar to the positive datasets but using the set of 400 deceptive and 400 truthful negative opinions from Ott et al.'s corpora. Accordingly, we randomly selected 80 negative deceptive opinions and 80 negative truthful opinions to build the test set. Then, the remaining 640 negative opinions were used to build six training sets of different sizes and distributions. They contain 20, 40, 60, 80, 100 and 120 negative deceptive opinions (positive instances) respectively. In all cases it was used a set of 520 unlabeled instances containing a distribution of 320 negative truthful opinions and 200 negative deceptive opinions.

Datasets of mixed polarity: These datasets were built to analyse the role of polarity in the detection of opinion spam. They were mainly assembled by combining the positive and negative sets previously described. Therefore, we form a test set consisting of 160 deceptive and 160 truthful opinions, and using the remaining 1280 opinions we built six training sets containing 40, 80, 120, 160, 200 and 240 deceptive opinions respectively (half of them positive opinions and the other half negative). In all cases it was used a set of 1040 unlabeled instances containing a distribution of 640 truthful opinions and 400 deceptive opinions.

3.5. Experimental Evaluation

3.5.1. Experimental settings

Document preprocessing: We removed all punctuation marks and numerical symbols, i.e., we only considered alphabetic tokens. We maintained the stop words, and converted all words to lowercase letter. These operations were applied on both labeled and unlabeled documents.

Learning algorithms: We used the Naïve Bayes (NB) classifier for all the experiments. We employed the implementation by Weka (Hall et al., 2009), considering all words occurring more than once in the training set as features. For the reported experiments we applied a binary weighting scheme. Additionally, in Section 3.5.5, we report results from a SVM classifier considering word unigrams and bigrams as features as suggested by (Ott et al., 2011, 2013). For this experiment we also employed the SVM implementation by Weka using a linear kernel and default parameters.

Evaluation measure: The evaluation of the effectiveness of the proposed method was carried out by means of the macro average of the F_1 measure for both classes, deceptive and truthful opinions. As mentioned before, in all the experiments we report the average results on the five different examples for each subset configuration of the datasets. The F_1 measure for each opinion category O_i is computed as follows:

$$f - measure(O_i) = \frac{2 \times recall(O_i) \times precision(O_i)}{recall(O_i) + precision(O_i)} \tag{3.1}$$

$$recall(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ opinions\ of\ O_i} \tag{3.2}$$

$$precision(O_i) = \frac{number\ of\ correct\ predictions\ of\ O_i}{number\ of\ predictions\ as\ O_i} \tag{3.3}$$

Statistical comparison of methods: Following the recommendation by (Demšar, 2006), we used the Wilcoxon Signed Ranks Test for comparing our method against other classification approaches. For these comparisons, we considered a 95 % level of signifiance (i.e., $\alpha = 0.05$) and a null hypothesis that both algorithms perform equally well. It is important to mention that for comparing any two methods, we

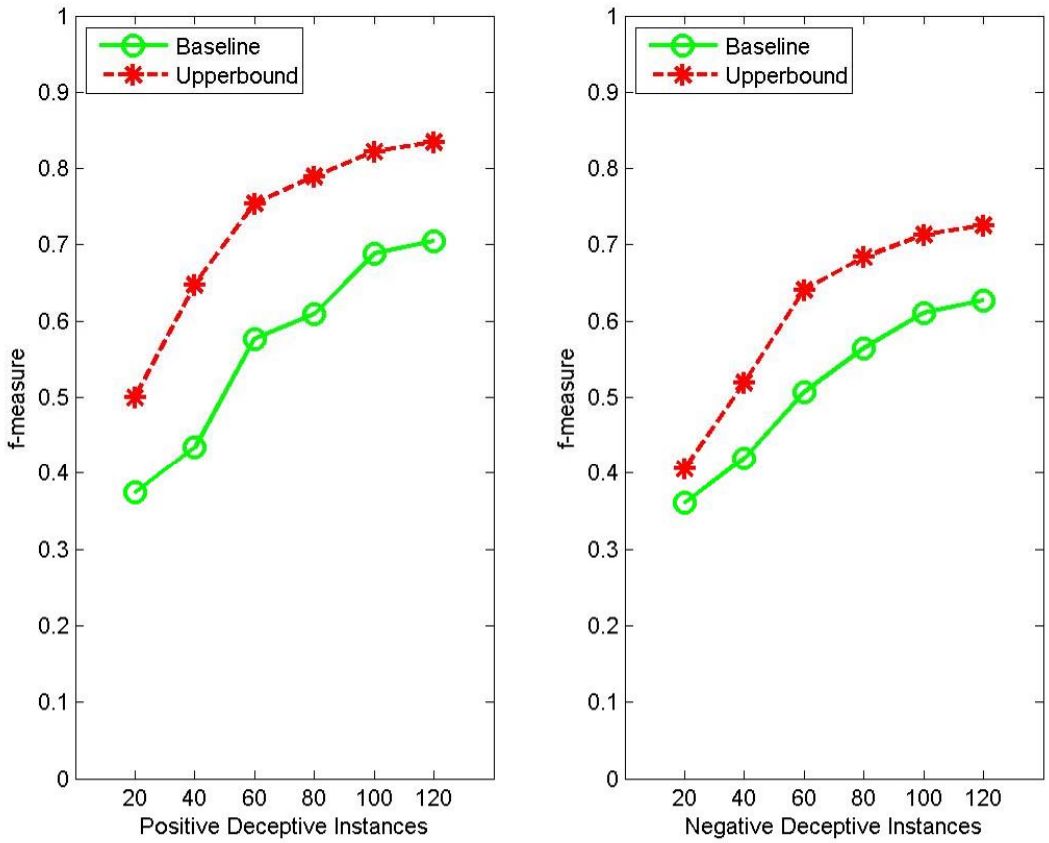

Figure 3.1: Baseline and Upperbound results for the different subsets of positive and negative opinions.

created two distributions with 20 values each, corresponding to their results in 5 folds from 4 collections (60, 80, 100 and 120 training instances).

3.5.2. Experiment 1: Lower and upper bounds for the PU-learning approach

This first experiment focused on evaluating the detection of positive and negative opinion spam under more realistic conditions, which consider only a few labeled deceptive opinions (and a set of unlabeled data) for training a classifier. The main objective of this experiment was to analyse the feasibility of the PU-learning approach for handling these complex but realistic scenarios.

This analysis was done using the first two datasets described in Section 3.4. As baseline we considered the results that were obtained by training a NB classifier

using the whole unlabeled set as the negative class[4]. This is a simple but common approach to build a binary classifier in case of lack of negative instances. It is worth mentioning that these results correspond to the results from the first iteration of the PU-learning approach. Moreover, as ideal performance of the PU-learning approach we considered the results that were obtained by training the NB classifier using only the truthful instances from the unlabeled set as the negative class. These results represent the upperbound for the proposed method since they could be reached only if the set of reliable negative instances is perfectly identified from the rest of the unlabeled instances. Figure 3.1 shows these two kinds of results for the different training subsets of datasets of positive and negative opinions.

Results from Figure 3.1 clearly indicate that classifying negative opinions is more difficult than the detection of positive deceptive and truthful opinions; the highest F_1 measure obtained for negative opinions was 0.74, whereas for positive opinions the ideal PU-learning approach could obtained a $F_1 = 0.85$. Furthermore, the improvement in the classification performance achieved by the PU-learning approach over the baseline was greater for positive opinions (30 %) than for negatives (19 %). This tendency confirm previous work's conclusions, which also suggest that negative spam is more complex for being identified.

Another interesting observation from Figure 3.1 is that PU-learning was incapable to learn a suitable classifier when having very few labeled deceptive opinions for training. Baseline results were lower than 0.5 when using 20 and 40 labeled examples, indicating that the initial selection of the reliable negative instances is very difficult under such circumstances. On the other hand, the upper-bound results were also not good; its poor performance could be attributed to two main reasons: the great imbalance in the training sets (20 or 40 deceptive opinions against 320 truthful opinions), and the difficulty of capturing the diversity in content and style of deceptive opinions from a small number of examples.

[4]Notice that in all our experiments the set of deceptive opinions are positive, negative or a combination of both, is used as the positive class.

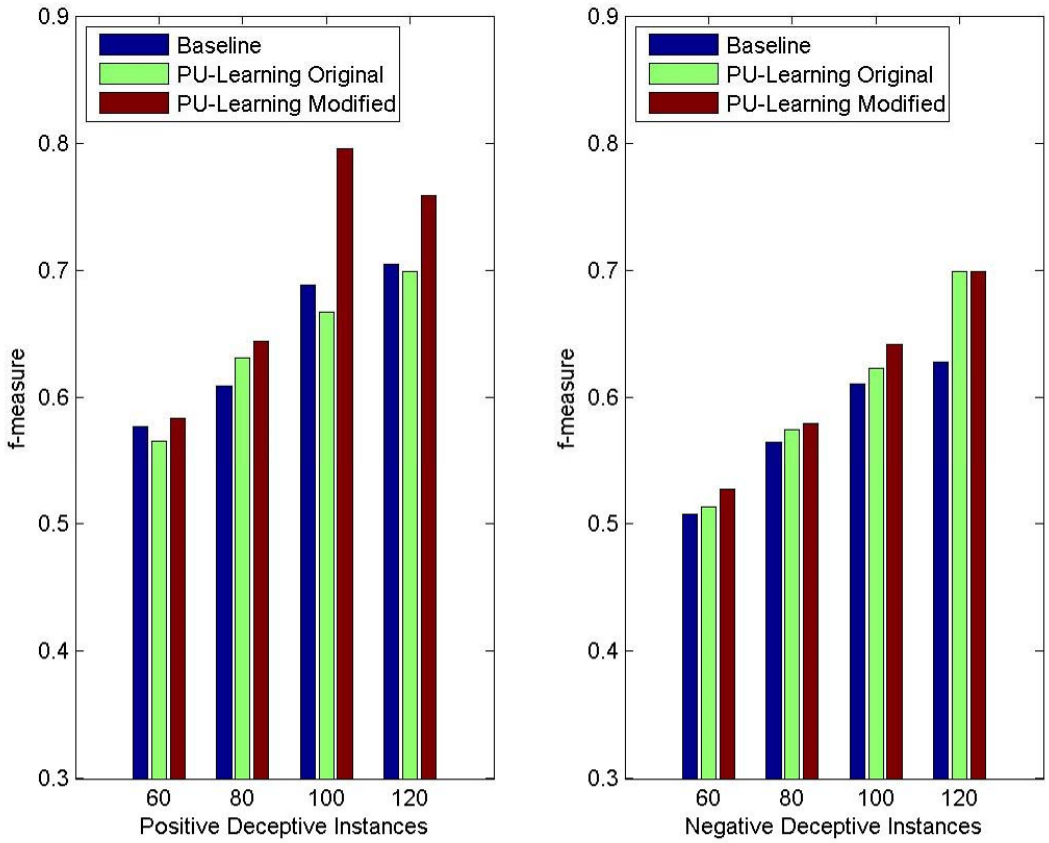

Figure 3.2: Results of the baseline, original PU-learning, and modified PU-learning in the classification of deceptive and truthful opinions from both polarities.

3.5.3. Experiment 2: Original vs modified PU-learning

This experiment focused on the comparison of the original and modified PU-learning methods in the classification of deceptive and truthful opinions. Figure 3.2 presents a general overview of the results obtained by these two approaches using training sets of positive and negative opinions of different sizes. These results show that the proposed PU-learning method *sistematically* outperformed baseline results as well as the results from the original PU-learning approach. In particular, it shows an average improvement of 8.2 % and 1.6 % over the original approach in the detection of positive and negative deceptive opinions respectively. Using the Wilcoxon test as explained in Section 3.5.1, we found that the proposed PU-learning approach is significantly better that both the baseline and original PU-learning method with $p < 0.05$ in both polarities.

Results from Figure 3.2 corroborate the already reported complexity involved in the classification of negative opinions; for this kind of opinions the best result of the proposed method was $F_1 = 0.7$ using 120 labeled deceptive opinions for training. In contrast, our method achieved a $F_1 = 0.79$ in the detection of positive deceptive and truthful opinions using only 100 labeled training samples. Searching for an explanation for this behavior, we noticed that the vocabulary employed in negative opinions was larger than the vocabulary from positives, indicating that their content is in general more detailed and diverse, and, therefore, that there larger training sets are needed for their adequate modelling.

Additional detailed results from this experiment are shown in Tables 3.1 and 3.2. These tables include the precision, recall and f-measure of the classification of deceptive as well as truthful opinions. They also show information about the number of iterations done by both PU-learning algorithms as well as the distribution of the training sets built by each of them.

Table 3.1: Detailed results on the classification of *positive* opinions using 60, 80, 100 and 120 labeled deceptive opinions (DP) and 520 of unlabeled examples (UN) for training. In this table, P, R and F state for precision, recall and f-measure respectively.

Initial Training Set	Used Method	Deceptive			Truthful			General F-measure	# of iterations	Final Training Set
		P	R	F	P	R	F			
60-DP/520-UN	BASELINE	0.896	0.268	0.408	0.605	0.975	0.746	0.577	1	60-DP/520-UN
	PU-L ORIGINAL	0.878	0.275	0.413	0.572	0.965	0.718	0.566	2	60-DP/473-UN
	PU-L MODIFIED	0.895	0.298	0.441	0.581	0.968	0.726	0.584	4	60-DP/394-UN
80-DP/520-UN	BASELINE	0.921	0.330	0.482	0.593	0.973	0.736	0.609	1	80-DP/520-UN
	PU-L ORIGINAL	0.925	0.363	0.519	0.604	0.970	0.744	0.632	2	80-DP/450-UN
	PU-L MODIFIED	0.842	0.415	0.547	0.618	0.933	0.742	0.645	7	80-DP/253-UN
100-DP/520-UN	BASELINE	0.919	0.408	0.561	0.621	0.965	0.756	0.689	1	100-DP/520-UN
	PU-L ORIGINAL	0.926	0.420	0.575	0.627	0.968	0.760	0.668	2	100-DP/432-UN
	PU-L MODIFIED	**0.852**	**0.728**	**0.780**	**0.768**	**0.868**	**0.811**	**0.796**	8	100-DP/112-UN
120-DP/520-UN	BASELINE	0.931	0.453	0.606	0.640	0.968	0.770	0.705	1	120-DP/520-UN
	PU-L ORIGINAL	0.916	0.480	0.626	0.648	0.955	0.772	0.699	2	120-DP/425-UN
	PU-L MODIFIED	0.803	0.700	0.743	0.738	0.823	0.774	0.759	7	120-DP/144-UN

In view that our main objective is the detection of opinion spam, it is of particular interest to analyse the classification results corresponding to the positive class (i.e., deceptive opinions). Table 3.1 shows a very good performance in the detection of positive deceptive opinions; whereas the original PU-learning approach obtained a maximum result of $F_1 = 0.626$, the proposed PU-learning method reached a $F_1 =$

0.78, giving an improvement of 24.6 %. Furthermore, this result presents a good trade-off between precision (0.85) and recall (0.72), compromise that could not be achieved by any of the other methods. On the other hand, as indicated in Table 3.2, the detection of negative deceptive opinions was not as good as in the case of positive opinions. The best result by the proposed method was $F_1 = 0.657$. However, the average improvement of the proposed method over the original PU-learning approach was of 11 % for all training conditions, indicating that the proposed approach is considerably better than the original one in the identification of opinion spam. It is worth mentioning that the better results by the proposed method in both polarities could be explained by its better selection of reliable negative instances. While the original approach retained more than 400 out of 500 instances in the negative class, our approach carried out a very hard selection of instances (i.e., truthful opinions), extracting in some cases less than 200 examples from the unlabeled set. Furthermore, the larger the set of labeled training instances, the higher the reduction made by the proposed method on the set of reliable negative instances. This is in contrast to the original PU-learning approach where the selection of reliable negative instances was uncorrelated with the number of labeled training instances.

Table 3.2: Detailed results on the classification of *negative* opinions using 60, 80, 100 and 120 labeled deceptive opinions (DP) and 520 of unlabeled examples (UN) for training. In this table, P, R and F state for precision, recall and f-measure respectively.

Training Set	Method Used	Deceptive			Truthful			F-measure general	# of iterations	Final training set
		P	R	F	P	R	F			
60-DN/520-UN	Baseline	0.906	0.178	0.312	0.548	0.980	0.703	0.508	1	60-DN/520-UN
	PU-L original	0.926	0.195	0.321	0.550	0.985	0.706	0.514	2	60-DN/483-UN
	PU-L modified	0.932	0.213	0.344	0.556	0.985	0.711	0.528	4	60-DN/404-UN
80-DN/520-UN	Baseline	0.903	0.270	0.412	0.570	0.968	0.717	0.565	1	80-DN/520-UN
	PU-L original	0.904	0.285	0.429	0.575	0.965	0.720	0.575	2	80-DN/464-UN
	PU-L modified	0.845	0.333	0.446	0.590	0.923	0.713	0.580	5	80-DN/295-UN
100-DN/520-UN	Baseline	0.926	0.333	0.486	0.593	0.970	0.736	0.611	1	100-DN/520-UN
	PU-L original	0.902	0.360	0.510	0.599	0.955	0.736	0.623	2	100-DN/450-UN
	PU-L modified	0.825	0.468	0.578	0.612	0.860	0.706	0.642	6	100-DN/202-UN
120-DN/520-UN	Baseline	0.898	0.370	0.517	0.604	0.953	0.738	0.628	1	120-DN/520-UN
	PU-L original	0.890	0.395	0.543	0.611	0.948	0.740	0.699	2	120-DN/438-UN
	PU-L modified	**0.788**	**0.595**	**0.657**	**0.672**	**0.803**	**0.723**	**0.699**	6	120-DN/177-UN

3.5.4. Experiment 3: Polarity and deception under PU-learning

The purpose of this experiment was to analyse the role of polarity in the classification of deceptive and truthful opinions, in the context of the proposed PU-learning method. To carry out this analysis we used the dataset of mixed polarity described in Section 3.4, and we evaluated the performance of two different classifier configurations. The first configuration considered *one* single classifier for detecting both positive and negative opinion spam. In other words, it did not take into account the polarity of reviews. In contrast, the second classifier configuration approached the identification of positive and negative spam as two different problems; it is mainly an *ensemble* of the two independent classifiers evaluated in the previous section. It is important to clarify that the first classifier used all available training data, whereas, in the ensemble configuration, each one of the classifiers was trained using only half of the data.

Table 3.3: Detecting Deceptive opinions when using 120, 160, 200 and 240 samples of Deceptive opinions and 1040 opinions of mixed polarities in the Unlabeled set (520 Deceptive and 520 Truthful).

Training Set	**Classifier Configuration**	**F-measure**		
		Deceptive Op	**Truthful Op**	**General**
120-D	One single Classifier	0.665	0.714	0.690
1040-U	Ensemble two classifiers	0.392	0.719	0.556
160-D	One single Classifier	0.740	0.797	0.769
1040-U	Ensemble two classifiers	0.496	0.727	0.612
200-D	One single Classifier	0.717	0.761	0.739
1040-U	Ensemble two classifiers	0.679	0.758	0.719
240-D	One single Classifier	0.771	0.790	0.781
1040-U	Ensemble two classifiers	0.700	0.748	0.724

Table 3.3 shows the results of this experiment. They indicate that for all the cases the configuration based on one classifier outperformed the results from the ensemble configuration; according to the Wilcoxon test, the one single classifier is statistically better than the ensemble in general F_1 measure with $p < 0.05$. It is worth noting that the advantage shown by the single classifier was particularly relevant for the cases using less training samples (120 and 160 mixed labeled deceptive opinions),

in which the improvement was around 25 %. These results are quite interesting and unexpected; they show that, despite their clear differences, positive and negative opinions have common elements that a classifier can exploit to enhance the spam modelling and classification. Moreover, the results indicate that in situations with lack of data, as the ones considered in this study, more data, even from a different polarity, it is always useful.

3.5.5. Experiment 4: On the choice of features and classifier

The goal of this last experiment was to evaluate the variation in the performance of the proposed PU-learning method when using other base classifier and a different set of features. Particularly, we employed a SVM classifier and combination of word unigrams and bigrams as features, such as considered by some previous successful works (Ott et al., 2011, 2013).

Table 3.4 shows the results of this experiment. According to the Wilcoxon Signed Ranks Test, these results indicate that the PU-learning method using NB as base classifier is significativelly better than its variant using the SVM classifier with $p < 0.05$, whatever the set of features was used. Somehow this conclusion was not completely unexpected; (Forman and Cohen, 2004) presented empirical evidence showing that Naïve Bayes models are often relatively insensitive to a shift in training distribution, and surpass SVM when there is a shortage of positives or negatives.

Regarding the used features results are not equally clear, the combination of unigrams and bigrams obtained better results than unigrams when using the NB classifier, but unigrams were the best features for the SVM classifier. For both configurations the differences in F_1 measure were statistically significant with $p < 0.05$. Although conclusions were slightly different for the two selected classifiers, it is important to point out that the proposed PU-learning method showed improvements to baseline results for the two polarities using any of the classifiers.

3.6. Conclusions and Future Work

Three are the contributions of this paper: (i) We approached the problem of the detection of deceptive opinions using the PU-learning technique because of the scar-

Table 3.4: Results of the classification of positive and negative opinion spam by Naïve Bayes (NB) and SVM using unigrams and bigramas as features. The values correspond to the F_1 measure for both classes, deceptive and truthful opinions.

Training Set	Corpus Used	Positive Opinions		Negative opinions	
		Unigrams	Uni+Bigrams	Unigrams	Uni+Bigrams
60-DN/520-UN	Baseline NB	0.577	0.604	0.508	0.579
	PU-L + NB	0.584	0.615	0.528	0.628
	Baseline SVM	0.419	0.344	0.420	0.341
	PU-L + SVM	0.443	0.360	0.433	0.344
80-DN/520-UN	Baseline NB	0.609	0.669	0.565	0.619
	PU-L + NB	0.645	0..686	0.580	0.649
	Baseline SVM	0.472	0.367	0.464	0.358
	PU-L + SVM	0.479	0.367	0.474	0.355
100-DN/520-UN	Baseline NB	0.689	0.691	0.611	0.650
	PU-L + NB	0.796	0.712	0.642	0.700
	Baseline SVM	0.502	0.406	0.503	0.379
	PU-L + SVM	0.539	0.410	0.524	0.387
120-DN/520-UN	Baseline NB	0.705	0.730	0.628	0.680
	PU-L + NB	0.759	0.778	0.699	0.727
	Baseline SVM	0.558	0.442	0.531	0.417
	PU-L + SVM	0.579	0.442	0.616	0.645

city of deceptive examples we believe it is the most adequate way; (ii) We proposed a novel, more conservative at the time of selecting the reliable negative examples, PU-learning approach; (iii) We analysed the role of the opinions' polarity in the detection of deception. The evaluation of the proposed method was carried out using the standard-de-facto hotel reviews dataset described in (Ott et al., 2013) that contains both *positive* and *negative* deceptive opinions. The results are encouraging and indicate that using only a hundred of examples of deceptive opinions for training it is possible to reach F_1 measures of 0.8 and 0.7 for positive and negative deceptive opinions respectively. They show the appropriateness of the proposed PU-learning conservative variant for detecting opinion spam, since its results consistently outperformed those obtained with the original approach in both kinds of deceptive opinions. In a further experiment where the role of opinions' polarity in the detection of deception is analysed, the obtained results confirm that negative deceptive opinions are more difficult to detect than positive ones, but they also show that having one single classifier for analysing both types of deceptive opinions is better that using two separate classifiers, suggesting that there are common characteristics in the way

people write positive and negative deceptive opinions.

As future work we aim at applying the novel PU-learning for detecting deceptive language to approach problems such as the detection of online sexual predators as well as the detection of lies in general.

Acknowledgments

This work is the result of the collaboration in the framework of the WIQEI IRSES project (Grant No. 269180) within the FP 7 Marie Curie. The work of the third author was in the framework the DIANA-APPLICATIONS-Finding Hidden Knowledge in Texts: Applications (TIN2012-38603-C02-01) project, and the VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

Capítulo 4

Detection of opinion spam with character n-grams

A continuación se presenta la versión de autor del artículo "Detection of opinion spam with character n-grams", publicado en la conferencia Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science, Volume 9042, Part II, páginas 285–294, 2015.

En este artículo se presentan los resultados obtenidos por el método propuesto de PU-Learning cuando se emplean como atributos los n-gramas de caracteres.*

Abstract

In this paper we presented a new approach to detect opinion spam using character n-grams as attributes, a powerful type of attributes exploited in the detection of mail spam, author attribution and others classification text tasks of Natural Language Processing. For the purpose of evaluation, we used a gold standard corpus composed of 1600 hotel reviews. We evaluated the positive, negative and full reviews. We compared the character n-gram against the word n-gram attributes by means of two experiments: in the first experiment we compared the effectiveness of the character n-gram attributes versus traditional approaches that uses word n-grams. In a second experiment we evaluated the effectiveness of this type of attributes decreasing the training set size. The results obtained show that character n-gram attributes have better robustness than word n-gram attributes, under this corpus type. The advantages of using character n-grams make it a good feature selection in the detection of opinion spam, because they obtained the content and writing style features better than word n-gram attributes. The best values obtained are a macro f-measure of *0.91* for the positive opinion spam and *0.87* in the case of negative opinion spam.

Keywords opinion spam, deceptive, character n-grams, word n-grams.

4.1. Introduction

In current times, the web users spend a lot of time reviewing opinions about products and/or services that are offered on the websites. These reviews may be positive or negative, that's it, in favour or against a product or service, but there is a special class of reviews, the *deceptive opinion spam*, which are fictitious opinions that have been deliberately written to sound authentic, in order to deceive the consumers. In a recent study shows that *80 %* of people are able to change their decision based on negative reviews about of any product or service.

The detection of deceptive opinion spam becomes an interesting problem, this text classification task has been studied using representations based in word n-grams and the best results were obtained with methods that also consider style attributes(Ott et al., 2011). The effectiveness obtained for human judges are almost *60 %*. The semi

supervised classifiers have proven to be more efficient with values of up to *85 %*; using word n-grams as attributes in conjunction with others stylometric features(Feng et al., 2012b; Ott et al., 2013).

We consider that detection of opinion spam task is closest to a stylistic classification because the deceptive and truthful opinions, in a particular domain, are similar in content but differ in the way, they communicate the message(Style). The character n-grams has been very effective in classification text tasks such as authorship attribution, spam mail, etc.(Feng et al., 2012a,b) because they capture information from the content and style, for example the character n-grams are able to get the emoticons {:-), ;-), :-(}, commonly used in social networks that reflects the polarity of the opinion, and some others combinations of characters harder to obtain when is used word n-grams(Stamatatos, 2012).

We proposed the use of character n-grams in detection of opinion spam. To do this we have two hypothesis: as first, we are going to test that the character n-grams representation is more appropriate than the word n-grams in the detection of opinion spam and, as second, we are going to show that the character n-grams are more robust than the word n-grams in scenarios where have only a few training data, due to their lower dispersion.

We made two experiments to demonstrate our hypothesis about the character n-grams attributes: In a first experiment we used a corpus containing 1600 hotel reviews. We analysed the performance for the positive, negative and the full reviews and, we obtained better results using character n-grams(The n values were 5,4 and 5 respectively) than word n-grams(Uni+Bigrams) as attributes.

Also, we realized a second experiment varying the training set size and holding the test set size, to demonstrate the robustness of the character n-grams attributes. The results obtained show that with a few data examples in the training set, is possible to get the character n-grams attributes with more information, without sacrifice the effectiveness.

The rest of the paper is organized as follows. Section 4.2 presents some related works in the field of opinion spam detection and the use of character n-grams as attributes for some classification text tasks. Section 4.3 presents the corpus used for experiments. Section 4.4 presents the experimental results and discusses its advan-

tages and disadvantages. Finally, Section 4.5 indicates the main contributions of the paper and provides some future work directions.

4.2. Related Work

The detection of spam has been traditionally studied in the context of e-mail (Drucker et al., 2002), and Web pages (Gyongyi et al., 2004; Ntoulas et al., 2006). For example, (Jindal and Liu., 2008) proposed detecting opinion spam by identifying duplicate content. In a subsequent paper, (Jindal et al., 2010) proposed to detect spammers by searching for unusual review patterns.

In this same category of unsupervised approaches, (Mukherjee et al., 2011) proposed a method for detecting groups of opinion spammers based on criteria such as the number of products for which the group work together and a high content similarity of their reviews. Similarly, in (Wu et al., 2010) the authors present a method to detect hotels which are more likely to be involved in spamming.

In the last tree years a lot of works have been realized in the detection of opinion spam (Feng et al., 2012a; Hernández et al., 2013; Liu., 2012), all of them used the word n-grams(Unigrams, Uni+Bigrams and Uni+Big+Trigrams) as the main attributes and, some of them added style writing attributes as LIWC (Language inquiry word count) (Ott et al., 2011), time Series Pattern (Lim et al., 2010; Sihong et al., 2012), statistical modelling language (Zhou et al., 2008) and syntactic stylometry (Feng et al., 2012b), these last as secondary attributes to improve the effectiveness.

We proposed the use of character n-grams in detection of opinion spam task, because these reviews have special characters, like emoticons and, these parts of the review reflect a writing style rather than content.

The character n-gram attributes were used in the detection of mail spam by (Kanaris et al., 2007) with a higher effectiveness than using word n-grams, the explanation of this improve it's because, using character n-grams, words like *f*r*e*e* are captured which are not obtained when used word n-grams.

Another interesting classification text task, where have been used the character n-grams, is the authorship attribution, works by (Stamatatos, 2012), shows better performance of this type of attributes in this task.

In the work (Blamey et al., 2012), they used the character n-gram attributes for sentiment classification in On-line Social Networks, and compares the performance versus the use of word n-grams and find a little improvement.

To the best of our knowledge, this work is the first in the detection of opinion spam, that uses the character n-grams attributes and, get some improvements versus the use of word n-grams.

4.3. Experimental setup

To test our two hypothesis we used a gold standard corpus of 1600 hotel reviews, this corpus was facilitated by Myle Ott[1]. These reviews are from 20 hotels in the down-town area of Chicago, each hotel has 80 reviews, half of those reviews are positive and the other half are negative, and finally we have 20 deceptive reviews and 20 truthful reviews of each one polarity.

The representation used in the two experiments was Bag of character n-grams (BOC) and Bag of Words n-grams (BOW). For the reported experiments we applied a binary weighting scheme. Additionally in the case of word n-grams, we made a preprocessing removing all punctuation marks and numerical symbols, i.e., we only considered alphabetic tokens. We maintained the stop words, and converted all words to lowercase letter.

We used the Naïve Bayes (NB) classifier for all the experiments and, we employed the implementation by Weka (Hall et al., 2009), considering all words occurring more than once in the training set as features.

For the evaluation measurements we used the three common parameters: *precision, recall and f-measure*, for each class (*Deceptive and Truthful*) of opinion spam. We included the *macro f-measure* to compare the effectiveness of the character n-grams versus the word n-grams in both experiments.

We made the significance test using the WILCOXON[2] signed rank test calculator. For these comparisons, we considered a 95 % level of significance (i.e., $\alpha = 0.05$) and a null hypothesis that both algorithms perform equally well, to demonstrate if the results obtained are statistical significant.

[1] http://myleott.com/op_spam

[2] http://www.socscistatistics.com/tests/signedranks/

4.4. Experiments

In this section we are going to present the results obtained for the two experiments realized to demonstrate our hypothesis. As a first experiment we compared the evaluation measurements obtained for character n-grams versus word n-grams. In the second experiment we are going to test the robustness of character n-grams when a few examples of deceptive opinion spam are present in the training set.

4.4.1. Experiment 1: Character vs. word n-grams

In this experiment, the objective is demonstrate that character n-grams are more appropriate than word n-grams to represent the content and writing style of opinion spam.

We separated the main corpus in three sub-corpus for our tests, first we put all the positive reviews in a sub-corpus called *Positive*, second we formed a sub-corpus with all the negative reviews that we called *Negative* and, the third sub-corpus was formed with all the 1600 reviews, grouped by classes(800 Deceptive and 800 Truthful reviews); this last corpus was called *full*. The following step was obtained the character n-gram and word n-gram attributes for each sub-corpus, to determine which are the best combination and later compare them using the evaluation measurements described in the section 4.3. With these three different sub-corpus we used the Naïve Bayes (NB) approach and made a ten fold cross validation (CV10), using the defaults values in Weka. The results are discussed in the next paragraphs.

Table 4.1: Results obtained with word ngrams and character n-grams for positive opinion spam.

ATTRIBUTES		**DECEPTIVE**			**TRUTHFUL**			**GENERAL**
type	*number*	*p*	*r*	*f*	*p*	*r*	*f*	*f*
unigrams	5920	0.886	0.873	0.879	0.874	0.888	0.881	0.880
uni+bigrams	44268	0.871	0.898	0.884	0.894	0.868	0.881	0.882
uni+big+trigrams	115784	0.867	0.900	0.883	0.896	0.863	0.879	0.881
character3	17182	0.846	0.865	0.855	0.862	0.843	0.852	0.854
character4	51105	0.894	0.878	0.884	0.879	0.893	0.886	0.885
character5	**108774**	**0.891**	**0.918**	**0.904**	**0.915**	**0.888**	**0.901**	**0.902**

As can be observed in table 4.1, the results obtained for the character n-grams are better than results for word n-grams (The best values are in bold). For the positive opinions set, the result shows that the best has a *macro f-measure* value of *0.902*, which correspond to the case *character5* versus the value of *0.882* obtained with the use of *uni+bigrams*. These values obtained means an improvement of 2.3 % in the *f-measure* for the Deceptive class, and an improvement of 2.3 % for the *macro f-measure*. For the negative opinions set, shows in table 4.2, the best result has a *macro f-measure* value of *0.872* with the use of *character4* versus the value of *0.854* obtained using *uni+bigrams*. These values obtained means an improvement of 2.0 % in the *f-measure* for the Deceptive class, and an improvement of 2.3 % for the *macro f-measure*.

Table 4.2: Results obtained with word ngrams and character n-grams for negative opinion spam.

ATTRIBUTES		**DECEPTIVE**			**TRUTHFUL**			**GENERAL**
type	*number*	*p*	*r*	*f*	*p*	*r*	*f*	*f*
unigrams	8131	0.861	0.835	0.848	0.840	0.865	0.852	0.850
uni+bigrams	65188	0.835	0.883	0.858	0.875	0.825	0.849	0.854
uni+big+trigrams	174016	0.808	0.893	0.848	0.880	0.788	0.831	0.840
character3	11942	0.822	0.923	0.869	0.912	0.800	0.852	0.861
character4	**37846**	**0.850**	**0.905**	**0.877**	**0.898**	**0.840**	**0.868**	**0.872**
character5	86248	0.841	0.870	0.855	0.865	0.835	0.850	0.852

Table 4.3: Results obtained with word ngrams and character n-grams for the full set of opinion spam.

ATTRIBUTES		**DECEPTIVE**			**TRUTHFUL**			**GENERAL**
type	*number*	*p*	*r*	*f*	*p*	*r*	*f*	*f*
unigrams	10431	0.871	0.854	0.862	0.857	0.874	0.865	0.864
uni+bigrams	93094	0.851	0.896	0.873	0.890	0.843	0.866	0.869
uni+big+trigrams	263133	0.835	0.906	0.869	0.898	0.821	0.858	0.864
character3	14338	0.812	0.951	0.876	0.941	0.788	0.853	0.865
character4	**47703**	**0.820**	**0.961**	**0.885**	**0.953**	**0.788**	**0.863**	**0.874**
character5	112066	0.805	0.968	0.879	0.959	0.765	0.851	0.865

For the full opinions set, involving positive and negative reviews, that is 800 deceptive opinions and 800 truthful opinions, the results showed in table 4.3 have

a *macro f-measure* value of *0.874* for *character4* versus a value of *0.869* obtained with the use of *uni+bigrams*. These values obtained means an improvement of 7.3 % in the *f-measure* for the Deceptive class, and an improvement of 1.0 % for the *macro f-measure*.

Another interesting point to observe in the tables 4.1, 4.2 and 4.3, is the column named *number*, just below the `ATTRIBUTES` top column, while in the rows for word n-grams the increasing number is exponential, for the character n-gram rows the increasing are less. Also can be mentioned that this number is always greater for the negative opinions. The explanation of this phenomenon is because the people who write a negative opinion, generally are most descriptive than people who write a positive opinion, because the number of words employed in the negative opinion is greater than positive opinion.

We obtained the 20 more important character n-grams with the Information Gain function of WEKA and we compared for the Positives and Negatives opinion spam. The results can be observed in table 4.4. In both cases the character n-grams are ordered according to their information gain value from the greatest to the least, we can see some character n-grams were the same for the positive and negative opinion spam. That is, these character n-grams appears in both type of opinion spam, then although this character n-gram were significant, they don't provide any information to assign the class to the opinion spam. Also we can note that the word *Chicago* appears in the first rows with the combinations of 4 and 5 character n-grams indicating that is most common in the opinion spam, without take account of the polarity or class of the opinion. There are some character n-grams ranked highest in the positive opinions than negative opinions, like *luxur*, the explanation that we can attribute to this behaviour is to the polarity's opinion.

4.4.2. Experiment 2: Character n-grams robustness

The second experiment has the objective of demonstrate the robustness of the character n-grams changing the training set size.

In this case we split the two sub-corpus formed for the positive and negative

Table 4.4: The 20 character n-grams with highest Information Gain values for postivie and negative opinions.

RANK	**POSITIVE**	**NEGATIVE**
1	ocati	Chic
2	chic	hica
3	icago	cago
4	hicag	icag
5	chica	Chi
6	luxur	ago
7	luxu	Hote
8	floo	Hot
9	floor	n Ch
10	hroom	luxu
11	throo	uxur
12	block	mell
13	bloc	smel
14	bath	sme
15	locat	l Ch
16	an av	lux
17	bathr	xury
18	athro	en I
19	cago	evat
20	n ave	leva

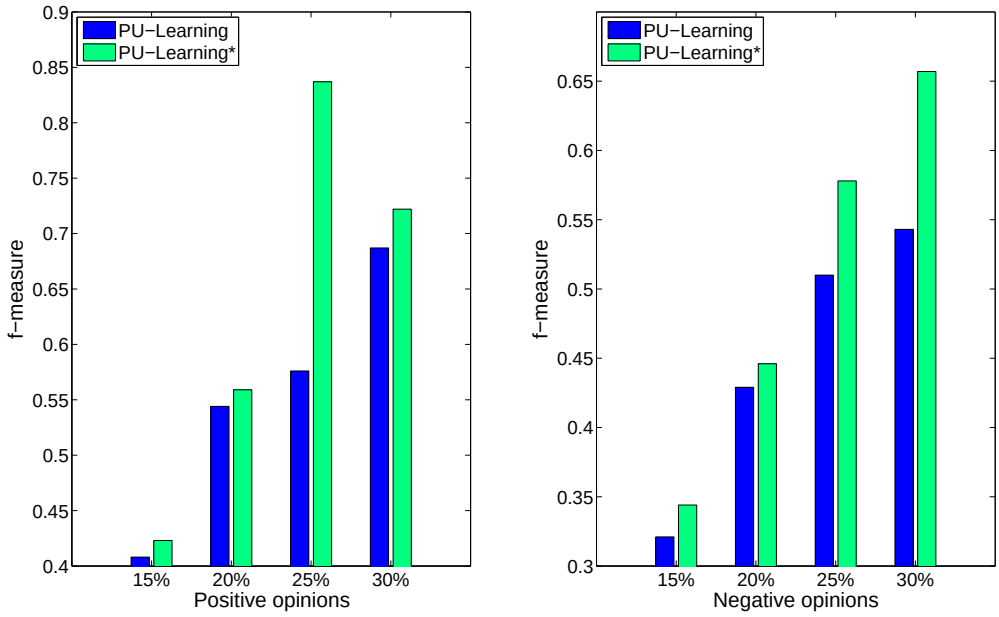

Figure 4.1: The macro f-measure variation with the training set size.

reviews, in four sub-corpus called *25 %, 50 %, 75 %* and *100 %*; each one of them containing ten folds to simulate the CV10 used in the Weka program for the first experiment. Then later, we plotted the results obtained of the evaluation measurement *f-measure* for the two types of attributes, the figure 4.1 shows the values when the number of instances in the training set were decreasing and the test set was maintained constant in each case.

From figure 4.1, we can observe the performance for the character n-grams and the word n-grams attributes, when the number of instances in the training set were decreasing, holding the number of instances in the test set, as were explained before. In these cases we obtained a maximum f-measure of *0.913* and *0.872* for the positive and negative opinions respectively. This behaviour indicates that the number of attributes, when was used the *100 %* of the training set was greater than really needed to get captured the writing style. The value of the *macro f-measure* has a maximum when we used only the *75 %* of the training set of the corpus.

We made the WILCOXON statistical significance test for the values represented

in the figure 4.1 and we obtained a consistently significance of the character n-grams over the word n-grams, even considering the case 25 % of positive opinions, where the *macro f-measure* of Uni+Bigrams is *0.875* and character n-grams is *0.872*, indicating for this particular combination than word n-grams were better than character5 n-grams.

4.5. Conclusions and future work

The conclusions that can be extracted from experiments realized and results obtained, are the following:

i) The character n-grams have the advantage of capture some features of the content and writing style, allowing to improve the effectiveness in the detection of opinion spam.

ii) There are some special characters, like emoticons, used in the opinion spam, than can be represented using the character n-grams and can't be with word n-grams.

iii) The positive opinion spam is represented better using character5 n-grams while negative opinion spam is represented with character4 n-grams.

iv) The character n-grams have a better robustness than word n-grams, when the train set size is small.

As future work, we are planning to introduce some heuristic functions to combine the character n-grams with word n-grams representations, as the following: Add the word and character n-grams attributes, intersection of both type of attributes and, a combination of character within word n-grams. We want to test these combinations in the detection of opinion spam, mail spam and authorship attribution classification text tasks.

Acknowledgments

This work is the result of the collaboration in the framework of the WIQEI IRSES project (Grant No. 269180) within the FP 7 Marie Curie. The work of the last author was in the framework the DIANA-APPLICATIONS-Finding Hidden Knowledge in

Texts: Applications (TIN2012-38603-C02-01) project, and the VLC/CAMPUS Microcluster on Multimodal Interaction in Intelligent Systems.

Capítulo 5

Discusión de los resultados

En este capítulo se presenta un resumen de los experimentos diseñados para la evaluación del método de PU-Learning, de los corpus empleados y de los resultados obtenidos, ya descritos de una manera más amplia en el Capítulo 2, en el Capítulo 3 y en el Capítulo 4. En el apartado 5.1 se presenta una breve descripción de los experimentos realizados para la evaluación del método de PU-Learning*. En el apartado 5.2 se presentan las características del corpus empleado en la realización de los experimentos. En el apartado 5.3 se presentan los resultados de la evaluación del método de PU-Learning* en la detección de opiniones falsas, particularmente en escenarios que consideran conjuntos pequeños de opiniones etiquetadas. En el apartado 5.4 se describen los resultados correspondientes al experimento donde se emplean los n-gramas de caracteres como atributos. Básicamente se presenta una comparación de la eficacia del método de PU-learning* usando las representaciones basadas en n-gramas de palabras y de caracteres. En el apartado 5.5 se comparan los resultados del método de PU-Learning* contra otros métodos de detección de opiniones falsas. Por último, en el apartado 5.6 se detallan los resultados de unos nuevos experimentos en un escenario de clasificación de dominios cruzados, donde no se dispone de instancias de entrenamiento en el dominio objetivo y se usan algunas provenientes de un dominio cercano.*

5.1. Evaluación experimental

Para realizar la evaluación del método de PU-Learning* en la detección de opiniones falsas, así como el uso de los n-gramas de caracteres como atributos, se diseñaron los siguientes experimentos:

- En un primer experimento se evaluó la eficacia del método de PU-Learning* utilizando como atributos los *n-gramas de palabras*. Se realizaron pruebas variando el tamaño del conjunto de entrenamiento para probar la robustez del método propuesto. Este experimento se diseñó de esta manera, porque muchas veces no se dispone de un número suficiente de opiniones etiquetadas para realizar el entrenamiento del clasificador.

- En el segundo experimento se evaluó la eficacia del método de PU-learning* utilizando como atributos los *n-gramas de caracteres*. El propósito general de este experimento fue evaluar la contribución de la información de estilo en la detección de opiniones falsas. Nuevamente se realizaron pruebas variando el tamaño del conjunto de entrenamiento para probar la robustez del método PU-Learning*.

- En el tercer y último experimento se evaluó la eficacia del método de PU-Learning* utilizando dominios diferentes para los conjuntos de entrenamiento y prueba. Esto es conocido como *dominios cruzados*. Nuevamente se utilizaron como atributos los n-gramas de caracteres. Con este tercer experimento se evalúa el desempeño del método de PU-Learning*, en escenarios donde no se cuenta con opiniones etiquetadas de algún dominio en particular, pero si se tienen opiniones etiquetadas de otro dominio diferente.

A continuación, en el siguiente apartado se presenta la configuración empleada en los experimentos planteados anteriormente.

5.2. Configuración experimental

La configuración de los experimentos realizados está basada en parte por las configuraciones empleadas por (Ott et al., 2011) y (Feng et al., 2012b). De estos

trabajos se puede observar que los resultados reportados son obtenidos empleando como atributos los n-gramas de palabras *(unigramas, bigramas y trigramas)*. También de estos trabajos se puede ver que se emplean los métodos de aprendizaje de Naïve Bayes *(NB)* y máquinas de vectores de soporte *(SVM)*. Por lo que también se emplean estas aproximaciones y atributos para evaluar el método de PU-Learning* .

Para el segundo experimento se cambiaron los atributos a n-gramas de caracteres buscando capturar además del contenido, el estilo de escritura de los documentos que forman el corpus empleado.

En el tercer experimento se utilizan las configuraciones anteriores pero tomando el conjunto de entrenamiento y el de prueba en dominios diferentes. Esto es conocido como dominios cruzados y nos sirve para probar el método propuesto en dominios donde no se dispone de ejemplos etiquetados, pero sí de otro dominio diferente.

Las configuraciones empleadas en cada experimento fueron las siguientes:

- Primer experimento:

  a) Los atributos que se emplearon fueron los n-gramas de palabras. Se hicieron pruebas con unigramas, unigramas + bigramas y unigramas + bigramas + trigramas.

  b) El pesado que se utilizó fue el booleano.

  c) Las medidas de evaluación que se registraron fueron las más comunes que son la precisión, la cobertura y el f-measure para cada clase.

- Segundo experimento:

  a) Los atributos que se emplearon fueron los n-gramas de caracteres. Se hicieron pruebas con secuencias de 3, 4 y 5 caracteres. Según estudios realizados por (Stamatatos, 2012) son los adecuados para el idioma inglés.

  b) El pesado que se utilizó fue el booleano.

  c) Las medidas de evaluación que se registraron fueron las de precisión, cobertura y f-measure para cada clase.

- Tercer experimento:

a) Los atributos que se emplearon fueron los n-gramas de caracteres. Se hicieron pruebas con secuencias de 3, 4 y 5 caracteres.

b) El pesado que se utilizó fue el booleano.

c) Las medidas de evaluación que se registraron fueron las de precisión, cobertura y f-measure para cada clase.

d) El conjunto de entrenamiento que se utilizó pertenece a un dominio diferente al dominio del conjunto de prueba.

En el siguiente sub-apartado se describen las características del corpus utilizado.

5.2.1. Colecciones

El primer corpus se obtuvo de (Ott et al., 2011), está formado por opiniones de hoteles y se encuentra disponible en la página del primer autor[1]. El segundo corpus empleado se obtuvo de (Li et al., 2013)[2], está formado por opiniones de hoteles, restaurantes y médicos. También se encuentra disponible bajo pedido en la página del primer autor. Estos corpus se describen con mas detalle en los siguientes sub-apartados.

5.2.1.1. Opiniones de hoteles

El primer corpus formado por opiniones de hoteles tiene un total de 1600 opiniones sobre el servicio que ofrecen 20 hoteles del centro de la ciudad de Chicago. Las opiniones fueron recopiladas a partir de diferentes fuentes de información como son Tripadvisor[3], Yelp[4] y Amazon Mechanical Turk[5] (AMT).

El corpus de opiniones de hoteles es un conjunto que está dividido en 4 categorías, cada una de las cuales contiene 400 opiniones, siendo estas categorías las siguientes:

a) Opiniones falsas favorables.

[1] http://myleott.com/op_spam
[2] Disponible por petición en el e-mail: bdlijiwei@gmail.com
[3] http://tripadvisor.com
[4] http://yelp.com
[5] http://mturk.com

b) Opiniones verdaderas favorables.

c) Opiniones falsas desfavorables.

d) Opiniones verdaderas desfavorables.

El corpus fue dividido en varios sub-corpus para probar la eficacia del método propuesto frente al tamaño del conjunto de entrenamiento. La primera división se realizó tomando en cuenta la polaridad de las opiniones, ya sea solo opiniones favorables o desfavorables. La segunda división se realizó sin tomar en cuenta la polaridad de las opiniones, sino más bien su falsedad o veracidad, en este caso se tomaron 800 opiniones falsas y 800 verdaderas para formar los sub-corpus.

En los siguiente párrafos se detalla cómo se formaron los diferentes sub-corpus para la realización de los experimentos.

Opiniones favorables: Para las opiniones favorables se formaron varios sub-corpus. Primero que nada se separó una parte de las opiniones para formar el conjunto de prueba conteniendo 80 opiniones falsas y 80 opiniones verdaderas. Con las 640 opiniones restantes se formaron seis conjuntos de entrenamiento más pequeños y se realizó de acuerdo a la siguiente tabla:

Tabla 5.1: Sub-corpus de opiniones favorables.

Porcentaje	Opiniones falsas	Opiniones no-etiquetadas
5 %	20	520
10 %	40	520
15 %	60	520
20 %	80	520
25 %	100	520
30 %	120	520

Para cada uno de los seis casos que aparecen en la tabla 5.1, se formaron 10 combinaciones diferentes a partir del corpus original. El propósito de obtener estas 10 combinaciones fue el de realizar una simulación de la técnica de validación cruzada

(Elkan, 2012), con 10 divisiones (ten fold cross validation) y de esta manera darle una mayor validez a los resultados experimentales obtenidos. En todos los casos se reporta el valor promedio obtenido para los 10 sub-corpus empleados en cada uno de ellos. Estos resultados se describen en el sub-apartado 5.3.1.

Opiniones desfavorables: En este caso, al igual que en las opiniones favorables se formaron varios sub-corpus. Primero se separaron las opiniones que forman parte del conjunto de prueba, 80 opiniones falsas y 80 verdaderas. Con las 640 opiniones desfavorables restantes se formaron seis conjuntos de entrenamiento de acuerdo a la siguiente tabla:

Tabla 5.2: Sub-corpus de opiniones desfavorables.

Porcentaje	Opiniones falsas	Opiniones no-etiquetadas
5 %	20	520
10 %	40	520
15 %	60	520
20 %	80	520
25 %	100	520
30 %	120	520

Para cada uno de los seis casos de conjuntos de entrenamiento, que aparecen en la tabla 5.2, se formaron 10 combinaciones diferentes a partir del corpus original. Los resultados obtenidos aparecen con mas detalle en el sub-apartado 5.3.2.

Opiniones de ambas polaridades: En este caso se dividió el corpus completo de acuerdo a la falsedad o veracidad de las opiniones, sin tomar en cuenta la polaridad. Nuevamente primero se separaron las opiniones que forman parte del conjunto de prueba, formado por 160 opiniones falsas y 160 verdaderas. Después se formaron seis casos diferentes de sub-corpus y los conjuntos de entrenamiento se eligieron de acuerdo a la siguiente tabla:

Tabla 5.3: Sub-corpus de opiniones falsas de ambas polaridades.

Porcentaje	Opiniones falsas	Opiniones no-etiquetadas
5 %	40	1040
10 %	80	1040
15 %	120	1040
20 %	160	1040
25 %	200	1040
30 %	240	1040

Nuevamente para cada uno de los seis casos de la tabla 5.3 se tomaron 10 combinaciones diferentes a partir del corpus original y los resultados se reportan en el sub-apartado 5.3.3 con mayor detalle.

5.2.1.2. Opiniones multi-dominio

Para la realización de los experimentos de dominios cruzados, se requiere que el conjunto de entrenamiento pertenezca a un dominio diferente al conjunto de prueba. Es decir primero se obtiene un clasificador binario en el dominio de entrenamiento, por medio de la aplicación del método de PU-Learning* y después se prueba el clasificador obtenido con un conjunto de opiniones que pertenecen al otro dominio.

El corpus utilizado en los experimentos de dominios cruzados contiene opiniones acerca del servicio de hoteles, restaurantes y médicos. Los corpus que se utilizaron están formados por los conjuntos de opiniones mostrados en la tabla 5.4 y están distribuidos de la siguiente manera:

Tabla 5.4: Corpus de opiniones utilizados en los experimentos de dominios cruzados.

Dominio	falsas	verdaderas
Restaurantes	90	200
Médicos	50	200
Hoteles	400	400

Como se puede observar la cantidad de opiniones falsas en los dominios de restaurantes y médicos es asimétrica, esto es no está balanceada la cantidad de opiniones falsas con respecto a la cantidad de opiniones verdaderas. Esta característica de los dominios de restaurantes y médicos, representa otra dificultad más del método de PU-Learning* para la detección de las opiniones falsas.

Los resultados obtenidos para los diferentes experimentos que se realizaron con dominios cruzados, se presentan en el apartado 5.6.

En el siguiente sub-apartado se detalla el pre-procesamiento realizado a los datos de los diferentes corpus.

5.2.2. Pre-procesamiento

El pre-procesamiento realizado a los sub-corpus seleccionados para los experimentos, consistió en la transformación de las cadenas de caracteres que contienen los documentos, en una representación compatible con los algoritmos de aprendizaje. Esta representación es un mapeo de los documentos en una forma compacta, típicamente un vector con términos ponderados, es decir que lo que se obtiene es una representación matricial *(m renglones y n columnas)*, donde cada renglón de la matriz corresponde a un documento de la colección.

Para cada renglón de la matriz existe un número n de columnas que representan los diferentes atributos que son elegidos para los términos que se presentan en todos los documentos. El valor de cada elemento de la matriz se le conoce como peso y se asocia con la importancia que tiene ese término en el documento. Existen varias maneras de calcular el peso y la que se eligió fue la denominada pesado Booleano.

La matriz resultante para los corpus utilizados en las pruebas del método propuesto, contiene un número muy grande de columnas. Es decir que tiene una alta dimensionalidad debido a las características del lenguaje utilizado para expresar las opiniones. En nuestro caso fue posible realizar los experimentos con este tipo de matrices, pero en otros casos puede ser necesario reducir las dimensiones de la matriz aplicando algunas técnicas para la selección de los atributos, como puede ser la de ganancia de información (Lee and Lee, 2006).

En el siguiente sub-apartado se presentan los tipos de atributos empleados para los experimentos realizados.

5.2.3. Representaciones

Las representaciones usadas para la realización de los experimentos fueron dos: la primera fue la de n-gramas de palabras *(unigramas, bigramas y trigramas)* y la segunda empleada fue la de n-gramas de caracteres *(3-caracteres, 4-caracteres y 5-caracteres)*. Ambas representaciones se detallan en los siguientes sub-apartados.

Bolsas de n-gramas de palabras.- Los n-gramas de palabras es una representación que ha sido utilizada ampliamente en las tareas de categorización de texto. Dado que logra capturar el contenido de los documentos de una manera tal, que si vemos un grupo de determinadas palabras que se repiten en una categoría, nos sirven de base para establecerlas como atributos propios de esa clase de documentos. Por el otro lado si se observa que un grupo de palabras no se repiten en una determinada categoría de documentos, también se pueden utilizar para determinar cuales documentos no pertenecen a dicha categoría.

En los experimentos realizados, para algunos sub-corpus existen una cantidad muy grande de n-gramas de palabras, del orden de hasta 250000, cuando se toman los grupos de unigramas + bigramas + trigramas. Todos estos n-gramas de palabras son empleados en las diferentes opiniones escritas por los usuarios.

En la siguiente tabla se presentan algunos ejemplos de corpus con la cantidad correspondiente del número de n-gramas de palabras obtenidos.

Tabla 5.5: Bolsas de n-gramas de palabras.

Polaridad	Falsas	Verdaderas	Unigramas	Bigramas	Trigramas
favorables	400	400	6635	47721	121219
desfavorables	400	400	8946	69790	150783
ambas	800	800	11983	181014	276319

A partir de la tabla 5.5 es posible observar otra complejidad más del problema de la detección de opiniones falsas. Dada la variedad de atributos que existen, algunos algoritmos de aprendizaje permiten realizar una discriminación tomando como base los atributos de mayor frecuencia.

Bolsas de n-gramas de caracteres.- Este tipo de representación logra capturar el contenido de los documentos, pero además obtiene información acerca del estilo de escritura. Esto es posible ya que por ejemplo al ir capturando los n-gramas de caracteres se obtiene la terminación de las palabras (las últimas letras que forman los unigramas), con esta información se puede saber si están escritas en un tiempo presente o pasado, singular o plural, etc. Con esta información acerca del estilo de escritura, es posible lograr una mejor clasificación de los documentos.

En la siguiente tabla se presentan algunos ejemplos de corpus, con la cantidad correspondiente del número de n-gramas de caracteres obtenidos.

Tabla 5.6: Bolsas de n-gramas de caracteres.

Polaridad	Falsas	Verdaderas	3-caracteres	4-caracteres	5-caracteres
favorables	400	400	17182	51105	108774
desfavorables	400	400	19734	61920	139731
Ambas	800	800	14338	47703	112066

A partir de la tabla 5.6, es posible observar que la cantidad de n-gramas de 3 caracteres es siempre mayor comparada con la cantidad de unigramas, reportada en la tabla 5.5. Mientras que para el caso donde se consideran la cantidad de n-gramas de 5 caracteres, es siempre menor comparada contra la cantidad de unigramas+bigrams+trigramas. Esto sucede por que los n-gramas de caracteres capturan una mayor cantidad de información de los documentos, tanto del contenido como del estilo de escritura.

A continuación en el siguiente sub-apartado se presentan las medidas de evaluación empleadas en los experimentos.

5.2.4. Medidas de evaluación

Para medir el rendimiento de los clasificadores se han utilizado muchas medidas de evaluación. Cada una de las cuales ha sido diseñada para evaluar algún aspecto del desempeño del clasificador, en la tarea de asignar una categoría a cada documento.

Tras la aplicación del método propuesto en los sub-corpus seleccionados para cada uno de los experimentos, lo que se obtiene es un clasificador binario. Este clasificador puede ser evaluado teniendo en cuenta las siguientes 4 cantidades de interés para cada categoría:

- a - El número de documentos asignado correctamente a esta categoría.
- b - El número de documentos asignado incorrectamente a esta categoría.
- c - El número de documentos rechazado incorrectamente de esta categoría.
- d - El número de documentos rechazado correctamente de esta categoría.

A partir de estas cantidades se pueden definir las siguientes medidas de evaluación:

$$precision = \frac{a}{a+b} \tag{5.1}$$

$$cobertura = \frac{a}{a+c} \tag{5.2}$$

$$f - measure = \frac{2 * precision * cobertura}{precision + cobertura} \tag{5.3}$$

Para el propósito de evaluación del desempeño del clasificador obtenido, se utilizaron las medidas de *precisión*, *cobertura* y *f-measure*. Estas tres medidas de evaluación se calculan para las dos clases de opiniones *(falsas y verdaderas)* en todos los experimentos realizados.

En el siguiente sub-apartado se presenta una descripción de la prueba de significancia estadística empleada.

5.2.5. Análisis estadístico

Para efectuar el análisis estadístico de los resultados obtenidos, se utilizó la prueba de significancia estadística propuesta por Wilcoxon (Wilcoxon, 1945). La prueba de Wilcoxon, es una prueba no paramétrica diseñada para evaluar la diferencia entre dos

tratamientos o condiciones, en las que se correlacionan las muestras. En particular, es adecuada para la evaluación de los datos de un diseño de medidas repetidas.

Como se mencionó en el apartado 5.2 los resultados que se muestran en las tablas corresponden a los valores promedio de 10 casos diferentes. Para realizar la prueba de significancia estadística de Wilcoxon, se introdujeron los valores individuales de cada caso y se obtuvo por medio de tablas los valores de *W (suma de diferencias)* y *Z (suma de rangos)*. Estos valores nos indican si la mejoría obtenida, por la aplicación del método propuesto es estadísticamente significativa.

En los siguientes sub-apartados se presentan un resumen de los resultados obtenidos en la aplicación del método propuesto para la detección de opiniones falsas con los experimentos realizados.

5.3. PU-Learning* con n-gramas de palabras

Propósito.- El experimento con n-gramas de palabras fue realizado con el propósito de evaluar la aplicación del método de PU-Learning*, en la detección de opiniones falsas partiendo de un conjunto pequeño de opiniones etiquetadas. Este experimento se realizó con los conjuntos de opiniones favorables, desfavorables y ambas, para estudiar el comportamiento del método de PU-Learning* con respecto a la polaridad de las opiniones.

También se compararon los resultados obtenidos con el método de PU-Learning*, contra los obtenidos utilizando el método de PU-Learning. En esta comparación se utilizaron las medidas de evaluación descritas en el sub-apartado 5.2.4.

Los resultados obtenidos para cada grupo de opiniones se presentan a continuación.

Análisis.- En cada una de las dos tablas que aparecen a continuación se tiene como primera columna el número de instancias de entrenamiento que se emplearon para cada caso. En todos los casos el tamaño del conjunto de prueba es el mismo *(80 opiniones falsas y 80 verdaderas)*. A continuación en la segunda columna aparece el método empleado (PU-Learning o PU-Learning*), en ambos métodos se utilizó la aproximación de Naïve Bayes *(NB)*.

En las siguientes dos columnas se presentan los valores de las tres medidas de evaluación seleccionadas para medir el desempeño del clasificador *(precisión, cobertura y f-measure)*. Las columnas en las cuales se refleja el efecto de aplicar los métodos (PU-Learning o PU-Learning*), son las que corresponden a la precisión, la cobertura y el f-measure de la clase opiniones falsas, ya que este es el tipo de opiniones que nos interesa detectar.

En el siguiente sub-apartado se presentan los resultados obtenidos para las opiniones favorables.

5.3.1. Opiniones favorables

En este caso solo se consideran las opiniones favorables. Los resultados obtenidos tras la aplicación del método de PU-Learning* se muestran en la tabla 5.7. Como se mencionó en el sub-apartado anterior primero se aplicó el método de PU-Learning y después el método de PU-Learning*. Este experimento se realizó con la aproximación de Naïve Bayes *(NB)* y los resultados se muestran en la tabla 5.7.

Tabla 5.7: Resultados del experimento con opiniones favorables.

Conjunto de Entrenamiento	**Aproximación empleada**	**Verdaderas**			**Falsas**		
		p	**c**	**f**	**p**	**c**	**f**
60 falsas	PU-Learning	0.569	0.975	0.719	0.913	0.263	0.408
520 no-etiquetadas	PU-Learning*	0.574	0.975	0.722	0.917	0.275	0.423
80 falsas	PU-Learning	0.611	0.963	0.748	0.912	0.388	0.544
520 no-etiquetadas	PU-Learning*	0.615	0.938	0.743	0.868	0.413	0.559
100 falsas	PU-Learning	0.623	0.950	0.752	0.895	0.425	0.576
520 no-etiquetadas	PU-Learning*	0.882	0.750	0.811	**0.783**	**0.900**	**0.837**
120 falsas	PU-Learning	0.679	0.950	0.792	*0.917*	0.550	0.687
520 no-etiquetadas	PU-Learning*	0.708	0.850	0.773	0.813	0.650	0.722

Discusión.- Los resultados de la tabla 5.7 muestran que el método de PU-Learning* es capaz de detectar las opiniones falsas favorables, siempre y cuando se tenga al menos el 20 % *(80 opiniones etiquetadas como falsas)* en el conjunto de entrenamiento. Al tener una cantidad más pequeña de opiniones etiquetadas, entre el 5 % y el 15 % *(20, 40 y 60 opiniones falsas)* en el conjunto de entrenamiento, no se obtienen valores

significativos de f-measure, para la clase de opiniones falsas. Estos bajos resultados se deben en parte, a que la cantidad de instancias etiquetadas es muy pequeña y el método no alcanza a aprender lo suficiente, para poder ayudar en la detección de las opiniones falsas.

Al ir incrementando la cantidad de opiniones etiquetadas como falsas del conjunto de entrenamiento, a valores entre el 20 % y el 30 % *(80, 100 y 120 opiniones falsas)*. Se puede notar que el método propuesto logra mejores resultados, comparables con los métodos que utilizan el corpus completo de opiniones disponible*(400 opiniones falsas)*.

De los valores presentados en la tabla 5.7 es posible observar que para el caso donde se emplea solo el 25 % del corpus completo *(100 opiniones falsas)*, se obtiene un valor del f-measure de 0.837. Recordemos que este valor corresponde al promedio de los 10 casos que se emplearon para realizar cada uno los cuatro casos. Este valor de f-measure es el máximo que se obtiene, al aplicar el método de PU-Learning*.

En el siguiente sub-apartado se presentan los resultados obtenidos para el caso de las opiniones desfavorables.

5.3.2. Opiniones desfavorables

Para el caso donde solo se consideran las opiniones desfavorables, los resultados también muestran una mejora significativa cuando se aplica el método de PU-Learning* en la detección de opiniones falsas desfavorables. Nuevamente se observó que cuando se tienen grupos muy pequeños de opiniones etiquetadas en el conjunto de entrenamiento, los resultados son muy por debajo incluso del azar *(*0.500*)*. En la tabla 5.8 se muestran los resultados a partir del 15 % *(60 opiniones falsas)*, hasta el 30 % *(120 opiniones falsas)* en el conjunto de entrenamiento.

Tabla 5.8: Resultados del experimento con opiniones desfavorables.

Conjunto de	**aproximación**	**Verdaderas**			**Falsas**		
Entrenamiento	**Empleada**	**p**	**c**	**f**	**p**	**c**	**f**
60 falsas	PU-Learning	0.550	0.985	0.706	0.926	0.195	0.321
520 no-etiquetadas	PU-Learning*	0.556	0.985	0.711	0.932	0.213	0.344
80 falsas	PU-Learning	0.575	0.965	0.720	0.904	0.285	0.429
520 no-etiquetadas	PU-Learning*	0.590	0.923	0.713	0.845	0.333	0.446
100 falsas	PU-Learning	0.599	0.955	0.736	0.902	0.360	0.510
520 no-etiquetadas	PU-Learning*	0.612	0.860	0.706	0.825	0.468	0.578
120 falsas	PU-Learning	0.611	0.948	0.740	0.890	0.395	0.543
520 no-etiquetadas	PU-Learning*	0.672	0.803	0.723	**0.788**	**0.595**	**0.657**

Discusión.- Los resultados obtenidos para las opiniones desfavorables, muestran que la dificultad en la detección de opiniones falsas desfavorables es mayor que cuando se detectaron las opiniones falsas favorables. Al incrementar la cantidad de opiniones etiquetadas del 15 % al 30 %, el valor de f-measure que corresponde a la clase de opiniones falsas mejora considerablemente. Se puede notar que el método de PU-Learning* logra sus mejores resultados, para el caso donde se utiliza el 30 % de opiniones etiquetadas *(120)*. Estos resultados son comparables con métodos que utilizan por completo el corpus de opiniones disponible *(400 opiniones falsas desfavorables)*.

Ahora presentaremos los resultados alcanzados para el caso donde se utiliza el corpus completo, pero sin tomar en cuenta la polaridad de las opiniones.

5.3.3. Opiniones falsas de ambas polaridades: favorables y desfavorables

Para este caso se utilizó el corpus completo de opiniones formado de ambas polaridades (favorables y desfavorables), o sea un total de 800 opiniones falsas y 800 verdaderas. En todos los casos el tamaño del conjunto de prueba utilizado fue de 160 opiniones falsas y 160 verdaderas. Los resultados se muestran utilizando el método de PU-Learning* en combinación con la aproximación de NB. Empleando los por-

centajes de opiniones falsas etiquetadas, que corresponden al 15 %, 20 %, 25 % y 30 % *(120, 160, 200 y 240 opiniones falsas)* en el conjunto de entrenamiento. En la tabla 5.9 se presentan los valores obtenidos en la detección de opiniones falsas de ambas polaridades.

Tabla 5.9: Resultados del experimento de opiniones falsas de ambas polaridades.

Conjunto de	**Aproximación**	**Verdaderas**			**Falsas**		
Entrenamiento	**Empleada**	**p**	**c**	**f**	**p**	**c**	**f**
120 falsas	PU-Learning	0.589	0.975	0.734	0.927	0.319	0.474
1040 no-etiquetadas	PU-Learning*	0.658	0.781	0.714	0.731	0.594	0.655
160 falsas	PU-Learning	0.634	0.975	0.768	0.946	0.438	0.598
1040 no-etiquetadas	PU-Learning*	0.719	0.894	0.797	0.860	0.650	0.740
200 falsas	PU-Learning	0.658	0.950	0.777	0.910	0.506	0.651
1040 no-etiquetadas	PU-Learning*	0.706	0.825	0.761	0.789	0.656	0.717
240 falsas	PU-Learning	0.607	0.963	0.788	0.933	0.519	0.607
1040 no-etiquetadas	PU-Learning*	0.759	0.825	0.790	**0.808**	**0.738**	**0.771**

Discusión.- Los valores de la tabla 5.9 muestran que cuando se combinan las opiniones etiquetadas de ambas polaridades en el conjunto de entrenamiento, es posible detectar las opiniones falsas a partir del 20 % *(160 opiniones falsas)*. Los valores reportados del f-measure obtenido para la clase de opiniones falsas, muestra un crecimiento monótono llegando al máximo valor obtenido de 0.771, cuando se utiliza el 30 % de opiniones etiquetadas *(240)*. Este valor es comparable con otros métodos que utilizan todo el corpus *(800 opiniones falsas)*.

El máximo valor obtenido para el f-measure de 0.771, está por debajo del obtenido en la tabla 5.7, que es de 0.837. Por lo que podemos afirmar que en el caso de las opiniones favorables, es más efectivo tener en cuenta primero la polaridad de las opiniones y después realizar la detección de las opiniones falsas. No así en el caso de las opiniones desfavorables, donde el valor obtenido de 0.771, es mayor que el de la tabla 5.8 que es de 0.657.

Conclusiones.- A partir de los valores mostrados en las tablas 5.7, 5.8 y 5.9; es posible obtener las siguientes conclusiones:

- El método de PU-Learning* es capaz de detectar las opiniones falsas con cualquier polaridad, utilizando como atributos los n-gramas de palabras.
- En el caso de las opiniones falsas desfavorables, el método de PU-Learning* tiene una eficacia menor. Esto se debe a la diversidad y tamaño del vocabulario empleado en las opiniones desfavorables.
- La eficacia del método de PU-Learning* cuando se utilizan conjuntos de entrenamiento que contienen opiniones falsas de ambas polaridades, se ve disminuida por lo que es mejor tener en cuenta primero la polaridad de las opiniones. Esto se debe a la gran cantidad de n-gramas de palabras presentes en el vocabulario.
- El f-meausure del método de PU-Learning* siempre fue mayor que el correspondiente al método de PU-Learning, para la clase de las opiniones falsas.

En el siguiente sub-apartado se presenta una descripción del análisis de significancia estadística realizado a los resultados obtenidos en los experimentos.

5.3.4. Análisis de significancia estadística

Para realizar el análisis de significancia se tomaron los valores obtenidos de f-measure en las tablas 5.1 y 5.2, para la clase de las opiniones falsas. Con los valores correspondientes al 15 %, 20 %, 25 % y 30 %, se formó la gráfica que aparece en la figura 5.1. Solo se tomaron los valores para los casos donde se emplea el método de PU-Learning y el método de PU-Learning*.

A partir de la figura 5.1 es posible observar que en todos los casos se tienen mejores resultados con el método de PU-Learning* que cuando se emplea el método de PU-Learning. Esta diferencia es más notoria si contamos con un número mayor de opiniones etiquetadas, como lo es para los casos del 25 % y 30 %.

Introduciendo estos valores en la prueba de significancia estadística de Wilcoxon, recordemos que para cada uno de los casos se tienen 10 valores diferentes, se obtienen

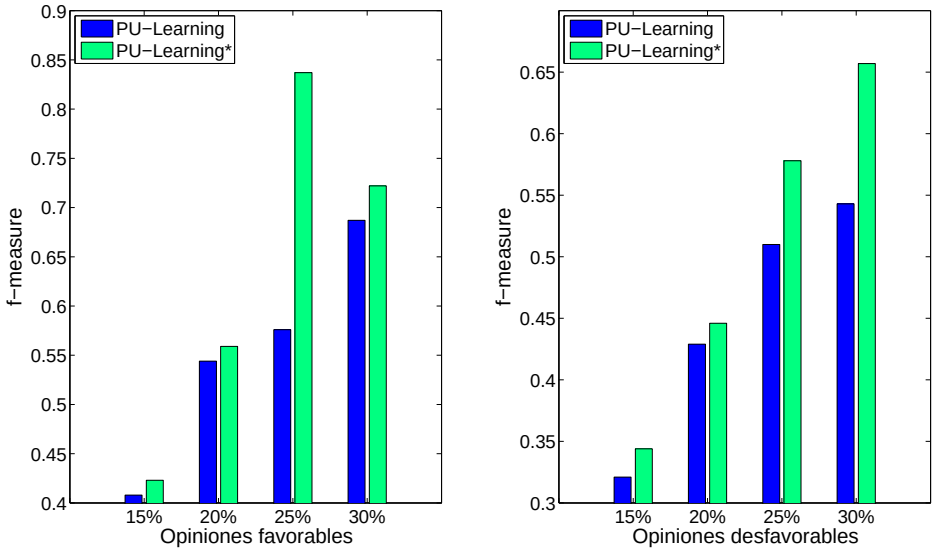

Figura 5.1: f-measure para diferentes tamaños del conjunto de entrenamiento.

los valores de la prueba y de acuerdo a las tablas que aparecen en la pagina web[6], nos muestran una significancia estadística. En otras palabras, con el empleo del método de PU-Learning* se obtuvieron resultados que son significativos estadísticamente.

Por estas razones podemos afirmar que los objetivos planteados, para el método de PU-Learninig*, se cumplen perfectamente para el dominio de opiniones del servicio que ofrecen los hoteles.

A continuación vamos a describir los resultados obtenidos para el segundo experimento realizado.

5.4. PU-Learning* con n-gramas de caracteres

Propósito.- El experimento se realizó con el propósito de combinar el método de PU-Learning* con los atributos de n-gramas de caracteres en la detección de opiniones falsas. Los resultados obtenidos con esta combinación, se comparan contra los resultados de las tablas anteriores 5.7 y 5.8, cuando se aplicaron los n-gramas de

[6]http://www.socscistatistics.com/tests/signedranks/Default2.aspx

palabras.

Los n-gramas de caracteres que se emplearon fueron 3, 4 y 5 caracteres, ya que estos son los apropiados para el lenguaje inglés, que es el idioma original de las opiniones analizadas.

Los resultados obtenidos para el f-measure promedio, aparecen en la figura 5.2 y después se aplica el análisis de significancia estadística de Wilcoxon, para determinar el comportamiento de la utilización de los n-gramas de caracteres contra los n-gramas de palabras.

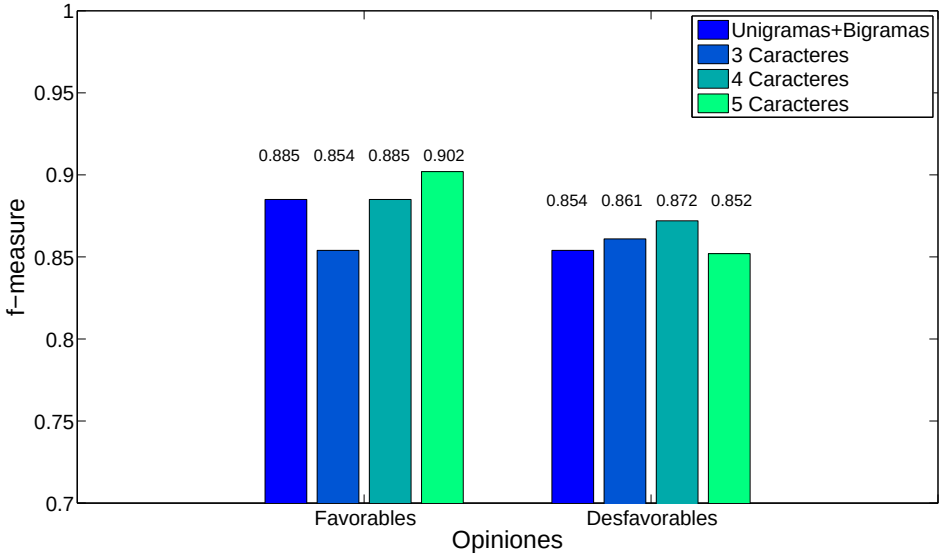

Figura 5.2: f-measure para diferentes tipos de atributos.

Análisis.- En la figura 5.2 se puede apreciar que para los casos de las opiniones favorables, el mejor resultado se consigue empleando n-gramas de 5 caracteres 0.902, mientras que cuando se utilizan n-gramas de palabras *(unigramas + bigramas)* se obtiene un f-measure de 0.885. Por otro lado para las opiniones desfavorables el mejor resultado se obtiene cuando se utilizan los n-gramas de 4 caracteres (0.872), el cual se compara contra 0.854, que es el mejor valor obtenido con n-gramas de palabras *(unigramas + bigramas)*.

Discusión.- Los resultados de la figura 5.2 nos muestran que en la detección de las opiniones falsas, ya sean favorables o desfavorables, es mejor emplear los n-gramas de caracteres como atributos. Esto se puede atribuir a que los n-gramas de caracteres logran capturar, tanto el contenido como el estilo de escritura de las opiniones falsas.

Conclusiones.- A partir de los valores del f-measure representados en la figura 5.2, es posible observar varias conclusiones dentro de las cuales podemos mencionar las siguientes:

- El método de PU-Learning* es más eficiente en la detección de opiniones falsas, utilizando como atributos los n-gramas de caracteres.
- La eficacia del método de PU-Learning* para las opiniones favorables, es mayor cuando se utilizan n-gramas de 5-caracteres.
- En el caso de las opiniones desfavorables, el método de PU-Learning* tiene una mejor eficacia cuando se utilizan n-gramas de 4-caracteres.
- En general podemos decir que los n-gramas de caracteres son una mejor opción para la representación de las opiniones (favorables y desfavorables).

En el siguiente apartado se presenta una comparación de los resultados alcanzados del método de PU-Learning* contra otros métodos que han sido aplicados al corpus de opiniones favorables de hoteles.

5.5. Comparación con otros métodos

A partir de los resultados obtenidos, se realizó una comparación con otros métodos que han utilizado el mismo corpus de opiniones falsas favorables y la técnica de validación cruzada para su entrenamiento. Estos métodos se agrupan de acuerdo al número de divisiones que se hayan empleado en la validación cruzada (5 ó 10).

Los otros métodos con los cuales se realizó la comparación, utilizan el corpus completo de opiniones favorables (400 falsas y 400 verdaderas), así como también

algunos tipos de atributos diferentes a los empleados por los métodos de PU-Learning y PU-Learning* que se incluyen en la comparación.

A continuación se muestran las características principales de las aproximaciones y atributos, empleados por los métodos con los cuales se realizó la comparación.

En el trabajo realizado por Ott et al. (2011) los autores presentaron por primera vez el corpus de opiniones favorables (falsas y verdaderas) acerca de hoteles. En el desarrollo de los experimentos de detección de opiniones falsas, los autores utilizaron como atributos los n-gramas de palabras combinado con LIWC de 80 atributos. El mejor resultado que obtuvieron fue empleando unigramas + bigramas + LIWC, usando un clasificador SVM y realizando una validación cruzada de 5 divisiones. En este caso obtuvieron un valor del f-measure de 0.898 para la clase de opiniones falsas favorables.

Feng and Hirst (2013) utilizaron las características de perfil de compatibilidad, combinado con los unigramas + bigramas. Además se propone un conjunto de reglas para la representación del corpus usando un clasificador SVM, también se realiza una validación cruzada de 5 divisiones. Con este tipo de representación obtienen un f-measure de 0.913. El mayor valor de los métodos contra los cuales se realizó la comparación.

Banerjee and Chua (2014) presentaron los resultados de un modelo de regresión logística utilizando 13 variables independientes diferentes, las cuales son: complejidad, legibilidad, adjetivo, artículo, sustantivo, preposición, adverbio, verbo, pronombre, pronombre personal, señales positivas, palabras de percepción y tiempo futuro. Se utilizó una validación cruzada de 10 divisiones y en este caso obtuvieron un f-measure de 0.705 para la clase de opiniones falsas favorables, el menor valor de los métodos que se compararon.

Ren et al. (2014) usaron un modelo semi-supervisado llamado población mezclada y de la propiedad individual. En este trabajo incorporan el método de PU–Learning y un clasificador SVM junto con una validación cruzada de 10 divisiones. En este caso obtuvieron un f-measure de 0.867.

Recientemente Cagnina and Rosso (2015) utilizaron una representación de atributos de baja dimensionalidad, basada en n-gramas de 4 caracteres dentro de unigramas (tokens), junto con información que proporciona el LIWC (pronombres, artículos y

verbos). El método obtuvo resultados comparables a los resultados obtenidos por el método de PU-Learning*, pero empleando un número de atributos mucho menor (1533 contra 60797 en el caso de opiniones favorables y 1497 contra 32063 en el caso de opiniones desfavorables).

A continuación se presentan las tablas comparativas con estos modelos, más los resultados obtenidos con la aplicación de la variante del método de PU-Learning (apartado 1.6.2) y el método de PU-Learning* (apartado 1.6.3), combinados con la representación de n-gramas de 5 caracteres para las opiniones favorables y de 4 caracteres en el caso de las opiniones desfavorables. Además se agregaron a la tabla los valores obtenidos cuando se emplearon los atributos de unigramas + bigramas. Estos valores fueron reportados en el Capítulo 2.

También se agregaron a las tablas los valores obtenido con el clasificador de una sola clase (one class SVM), el baseline (el valor inicial que se obtuvo antes de aplicar cualquiera de los métodos). En el último renglón de las tablas 5.10 y 5.11 aparece el límite superior (el valor que se obtiene cuando se emplea el corpus completo de opiniones) en combinación con la aproximación de NB y una validación cruzada de 10 divisiones.

Tabla 5.10: Comparación de los resultados obtenidos por el método de PU-Learning* contra otros métodos que emplean el mismo corpus de opiniones favorables.

	Modelo	Atributos	f-measure
5 folds cross-validation	Ott et al.	unigramas+bigramas+LIWC	0.898
	Feng y Hirst	unigramas+bigramas+perfil	0.913
10 folds cross-validation	Banerjee y Chua	regresión logística	0.705
	Ren et al.	mezcla y propiedad	0.867
	Cagnina y Rosso	n-gramas 4 caracteres en tokens	0.890
	one class SVM	unigramas+bigramas	0.465
	baseline	unigramas+bigramas	0.756
	PU-Learning	unigramas+bigramas	0.811
	PU-Learning*	unigramas+bigramas	0.885
	PU-Learning	n-gramas 5 caracteres	0.822
	PU-Learning*	n-gramas 5 caracteres	**0.902**
	upperbound	n-gramas 5 caracteres	0.905

A partir de la tabla 5.10 es posible observar que en una validación cruzada de 10 divisiones el mejor resultado es obtenido por el método de PU-Learning* con los

n-gramas de 5 caracteres y un clasificador NB (f-measure de 0.902). En la tabla 5.11 de las opiniones desfavorables, el mejor valor obtenido es de 0.872 cuando se utilizan los n-gramas de 4 caracteres.

Tabla 5.11: Comparación de los resultados obtenidos por el método de PU-Learning* con los métodos que emplean el mismo corpus de opiniones desfavorables.

	Modelo	Atributos	f-measure
5 folds cross-validation	Ott et al.	unigramas+bigramas+LIWC	0.860
10 folds cross-validation	Cagnina y Rosso	n-gramas 4 caracteres en tokens	0.865
	one class SVM	unigramas+bigramas	0.465
	baseline	unigramas+bigramas	0.595
	PU-Learning	unigramas+bigramas	0.623
	PU-Learning*	unigramas+bigramas	0.854
	PU-Learning	n-gramas 4 caracteres	0.802
	PU-Learning*	n-gramas 4 caracteres	**0.872**
	upperbound	n-gramas 4 caracteres	0.890

Estos valores indican que si es posible detectar las opiniones falsas utilizando el método de PU-Learning* y que los resultados son comparables con los otros métodos descritos.

La diferencia principal de aplicar el método de PU-learning*, radica en que solo se utiliza una pequeña cantidad de ejemplos etiquetados (100 opiniones falsas) para su implementación. Mientras que los otros métodos utilizan el corpus completo de opiniones (400 falsas y 400 verdaderas) etiquetadas.

Por otro lado también es posible observar, a partir de los resultados reportados en los últimos renglones de las tablas 5.10 y 5.11, la mejora que se obtiene cuando el método de PU-Learning* emplea como atributos, respectivamente, n-gramas de 5 y de 4 caracteres.

En el siguiente apartado se describen los resultados obtenidos al aplicar el método de PU-Learning* en un escenario de dominios cruzados.

5.6. PU-Learning* en dominios cruzados

Propósito.- El experimento de dominios cruzados se realizó con el propósito de aplicar el método de PU-Learning* empleando los n-gramas de palabras y de ca-

racteres como atributos, cuando no se dispone de opiniones falsas en un dominio específico y se usan opiniones falsas de un dominio cercano para el entrenamiento. En este experimento se realiza la aplicación del método de PU-Learning* en uno de los dominios disponibles *(hoteles)* y después se utiliza como dominio de prueba otro diferente *(restaurantes o médicos)*. Todas las opiniones empleadas en este experimento fueron del tipo favorables (falsas y verdaderas) debido a que no se cuenta con opiniones del tipo desfavorables para los dominios de restaurantes y médicos.

Análisis.- Los sub-corpus seleccionados para la aplicación del método de PU-Learning* corresponden al 25 %, 50 % y 75 % del corpus completo, estos sub-corpus quedaron formados de acuerdo a la tabla 5.12. Además se agregó a la tabla el renglón correspondiente al 100 % del corpus de opiniones del dominio de hoteles (en este caso no es posible aplicar el método de PU-Learning*, ya que no cuenta con un conjunto de opiniones no-etiquetadas).

Tabla 5.12: Sub-corpus utilizados para el conjunto de entrenamiento de los experimentos de dominios cruzados.

Porcentaje	Conjunto de Entrenamiento
25 %	100-falsas/700-no-etiquetadas
50 %	200-falsas/600-no-etiquetadas
75 %	300-falsas/500-no-etiquetadas
100 %	400-falsas/400-verdaderas

Para simular la técnica de validación cruzada con 10 divisiones, se realizaron 10 combinaciones para cada uno de los tres sub-corpus utilizados en los experimentos. En cada uno de los 10 casos se aplicó el método de aprendizaje de NB y se utilizaron como atributos los n-gramas de palabras y de caracteres. Los resultados que se presentan en las tablas 5.13 y 5.14 corresponden al promedio obtenido para los 10 casos diferentes.

Tabla 5.13: Resultados del experimento de dominios cruzados utilizando como conjunto de entrenamiento hoteles y como conjunto de prueba restaurantes.

Atributos	**25 %**	**50 %**	**75 %**	**100 %**
UNIGRAMAS	0.765	0.792	0.783	0.767
UNI+BIGRAMAS	0.717	0.792	0.781	0.769
3-CARACTERES	0.714	0.787	0.809	0.791
4-CARACTERES	0.763	0.810	0.832	0.832
5-CARACTERES	0.769	0.795	0.794	0.774

Tabla 5.14: Resultados del experimento de dominios cruzados utilizando como conjunto de entrenamiento hoteles y como conjunto de prueba médicos.

Atributos	**25 %**	**50 %**	**75 %**	**100 %**
UNIGRAMAS	0.586	0.510	0.485	0.396
UNI+BIGRAMAS	0.558	0.575	0.476	0.342
3-CARACTERES	0.518	0.570	0.584	0.472
4-CARACTERES	0.586	0.609	0.553	0.479
5-CARACTERES	0.636	0.582	0.560	0.441

Como se puede apreciar de la tabla 5.13 los mejores resultados se obtienen cuando se utiliza el 75 % del corpus de opiniones y se aplica el método de PU-Learning* usando como atributos los n-gramas de 4 caracteres.

En la tabla 5.14 se puede observar que para el caso de la combinación de dominios hoteles-médicos, el mejor resultado se obtiene usando el 25 % del corpus y como atributos los n-gramas de 5 caracteres.

Afinidad.- Para medir la afinidad entre dos dominios se puede emplear el índice de Jaccard (1901), este valor nos indica si un dominio presenta una cierta afinidad con respecto a otro, en el rango de 0 (no afines) a 1 (máxima afinidad). Para calcular el índice de Jaccard, primero se tiene que encontrar el número de atributos de cada uno de los dominios involucrados (denominados $Dominio_1$ y $Dominio_2$) y después el

número de atributos que son comunes a ambos dominios (denominado $Dominio_1 \cap Dominio_2$). Luego simplemente se efectúa la relación definida por la formula del índice de Jaccard representada en la siguiente ecuación:

$$índice\,de Jaccard = \frac{Dominio_1 \cap Dominio_2}{Dominio_1 + Dominio_2 - Dominio_1 \cap Dominio_2} \tag{5.4}$$

Los valores del índice de Jaccard que se obtuvieron para las combinaciones de dominios utilizados en los experimentos se muestran en las tablas 5.15 y 5.16. Los valores más altos del índice de Jaccard, indican que la afinidad entre los dominios de hoteles-restaurantes es alta (ver el caso de n-gramas de 3 caracteres). Por otro lado, los valores que se obtienen para el caso de los dominios de hoteles-médicos indican que presentan una menor afinidad en el lenguaje utilizado para la escritura de las opiniones.

Tabla 5.15: Índice Jaccard para los dominios de hoteles-restaurantes.

Atributos	**hoteles**	**restaurantes**	**hoteles ∩ restaurantes**	**índice Jaccard**
UNIGRAMAS	5920	3901	2226	0.293
UNI+BIGRAMAS	44268	23731	8128	0.135
3-CARACTERES	7963	6312	5084	0.553
4-CARACTERES	26210	19543	13918	0.437
5-CARACTERES	60797	41066	25498	0.333

Tabla 5.16: Índice Jaccard para los dominios de hoteles-médicos.

Atributos	**hoteles**	**médicos**	**hoteles ∩ médicos**	**índice Jaccard**
UNIGRAMAS	5920	2882	1596	0.221
UNI+BIGRAMAS	44268	17485	5470	0.097
3-CARACTERES	7963	5059	4196	0.475
4-CARACTERES	26210	14962	11118	0.369
5-CARACTERES	60797	30643	19075	0.274

Es posible observar a partir de las tablas 5.15 y 5.16, que los atributos que presentan un mejor valor del índice de Jaccard son los n-gramas de caracteres, indicando con esto que este tipo de atributos es más adecuado que los n-gramas de palabras, para la realización del experimento de dominios cruzados.

Discusión.- A partir de la tabla 5.13 es posible observar que cuando se utilizó la combinación de los dominios hoteles-restaurantes, se obtiene un valor de f-measure de 0.832, indicando que existe una alta afinidad entre los dominios. Además se puede observar a partir de la tabla 5.14 que los valores de f-measure obtenidos, para el caso donde se emplea el corpus de opiniones de médicos como conjunto de prueba, no son tan significativos ya que el mejor valor de f-measure obtenido es de 0.636, cuando se emplean como atributos los n-gramas de 5-caracteres.

Conclusiones.- A partir de los valores obtenidos en las tablas 5.13 y 5.14 es posible realizar varias conclusiones dentro de las cuales podemos mencionar las siguientes:

- El método de PU-Learning* es capaz de detectar las opiniones falsas en ambientes de dominios cruzados.
- El método de PU-Learning* obtiene mejores resultados cuando los dominios presentan una mayor afinidad en el lenguaje empleado en la escritura de las opiniones. Como es el caso de los dominios de hoteles y restaurantes.
- La eficacia del método de PU-Learning*, cuando se utilizan dominios que no presentan afinidad en el lenguaje empleado, se ve disminuida. Por lo cual es mejor buscar opiniones etiquetadas como falsas que si correspondan al dominio de prueba.
- El método de PU-Learning* obtuvo una eficiencia mejor cuando se emplearon n-gramas de 4-caracteres como atributos, tanto para dominios afines *(hoteles y restaurantes)*, como para dominios que presentan una afinidad menor *(hoteles y médicos)*.

En el siguiente capítulo se presentan las conclusiones obtenidas de este trabajo de tesis, así como las recomendaciones de algunos trabajos que se pueden realizar en el futuro y también la lista de publicaciones realizadas.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones de este trabajo de tesis, asimismo se enlistan algunas ideas para trabajo futuro. De manera particular en el apartado 6.1 se realiza una descripción de las principales contribuciones de esta investigación y se discuten algunas conclusiones obtenidas a partir de los resultados experimentales. En el apartado 6.2 se describen las líneas de investigación que se pueden continuar a partir del presente trabajo. Por último, en el apartado 6.3 se enlistan las publicaciones obtenidas a partir de este trabajo de tesis.

6.1. Conclusiones

Este trabajo de investigación se enfocó en la tarea de detección automática de opiniones falsas. A diferencia de los métodos existentes que han abordado esta tarea desde un enfoque supervisado, usando tanto opiniones falsas como verdaderas para entrenar sus modelos, en este trabajo se consideró un escenario más realista donde sólo se dispone de un puñado de opiniones falsas para construir los modelos de clasificación.

La principal contribución de este trabajo fue el diseño del método llamado PU-Learning*, que permite detectar opiniones falsas partiendo de un conjunto pequeño de instancias etiquetadas como opiniones falsas y otro conjunto más grande de instancias no-etiquetadas. Los resultados experimentales nos permiten concluir que sí es posible detectar de manera efectiva opiniones falsas usando el método propuesto.

Adicionalmente nuestros experimentos también nos permiten formular las siguientes conclusiones respecto al método PU-Learning*:

- Es más efectivo para esta tarea que el método PU-Learning original; esto se debe en gran manera al uso de criterios más conservadores y exigentes para la selección de las opiniones (presumiblemente) falsas del conjunto no-etiquetado.
- Es efectivo para la detección de opiniones falsas de ambas polaridades, sin embargo los resultados obtenidos fueron mejores en la detección de opiniones falsas favorables. Este comportamiento del método propuesto coincide con trabajos previos, y se origina en la mayor diversidad y mayor nivel de especificidad de las opiniones falsas desfavorables.
- Es una solución adecuada para la detección de opiniones falsas en escenarios de dominios cruzados, particularmente cuando los dominios fuente y objetivo son afines, es decir, cuando éstos muestran una alta intersección de su vocabulario.

Adicional al diseño del método PU-Learning*, una segunda contribución de este trabajo fue la propuesta de una representación de los documentos apropiada para esta tarea, que incorpora tanto aspectos de su contenido como de su estilo. Esta representación fue a través de n-gramas de caracteres. Los resultados obtenidos con

esta representación fueron superiores a los obtenidos con la representación tradicional de bolsa de palabras, permitiendo concluir que la información estilísitca es también importante para la detección de opiniones falsas. Los experimentos realizados también indican que para modelar adecuadamente el estilo de escritura de las opiniones falsas y verdaderas es necesario disponer de grandes conjuntos de entrenamiento; usando conjuntos pequeños las diferencias en los resultados de ambas representaciones fueron estadísticamente no significativas.

6.2. Trabajo futuro

A continuación se describen varias ideas para continuar con el trabajo de esta investigación. Para mayor claridad estas ideas se han agrupado en tres categorías: ideas relacionadas con la mejora del método PU-learning*, ideas relacionadas con la representación de los documentos, e ideas de aplicación del enfoque propuesto en tareas afines.

Ideas para mejorar el método PU-learning:*

- Proponer una variante del método que no solo extraiga opiniones (presumiblemente) verdaderas del conjunto de opiniones no etiquetadas, sino que también considere la extracción de opiniones (presumiblemente) falsas, y su incorporación iterativa y gradual al conjunto de entrenamiento original.
- Modificar las condiciones de paro del método propuesto, con el propósito de reducir el número de iteraciones y con ello mejorar su eficiencia.

Ideas relacionadas con la representación de los documentos:

- Analizar la contribución de cada tipo de n-grama de caracteres en la detección de opiniones falsas. Es sabido que existen distintos tipos de n-gramas de caracteres, por ejemplo algunos capturan las raíces de las palabras y otros sus terminaciones (y por tanto uso de comparativos, superlativos y tiempos verbales, entre otros fenómenos), y es por ello importante determinar el tipo de información que es mas relevante para la detección de opiniones falsas.
- Evaluar otros tipos de atributos estilísticos en la detección de opiniones falsas. De manera particular se sugiere la evaluación de atributos usados en la tarea de atribución de autoría y detección de plagio.

Ideas relacionadas con la aplicación del método propuesto:

- Probar el método propuesto PU-learning* en otras tareas afines tal como la verificación de autoría, donde la obtención o definición de ejemplos negativos de entrenamiento es complicada o incluso imposible.

6.3. Publicaciones

A continuación se presenta una lista de las publicaciones realizadas durante el desarrollo de este trabajo de tesis:

a) E. Ferretti, D. H. Fusilier, R. Guzmán-Cabrera, M. Montes-y-Gómez, M. Errecalde y P. Rosso. ***On the use of PU Learning for quality flaw prediction in Wikipedia.*** CLEF 2012 Evaluation Labs and Workshop, On line Working Notes, 2012, Roma, Italia, Septiembre 17-20.

b) D. Hernández, R. Guzmán, M. Montes-y-Gómez y P. Rosso. ***Using PU-learning to detect deceptive opinion spam.*** Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis for Computational Linguistics, NAACL, páginas 38–45, 2013, Atlanta, USA, Junio 14.

c) D. Hernández, M. Montes-y-Gómez, P. Rosso y R. Guzmán. ***Detecting positive and negative deceptive opinions using PU-learning.*** Information Processing & Management, Volumen 51, Número 4, páginas 433–443, 2015.

d) D. Hernández, M. Montes-y-Gómez, P. Rosso y R. Guzmán. ***Detection of opinion spam with character n-grams.*** Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science, CICLing-2015, Springer-Verlag, Volumen 9042, Parte II, páginas 285-294, 2015.

Bibliografía

S. Banerjee and A. Y. K. Chua. Dissecting genuine and deceptive kudos: The case of online hotel reviews. In *International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Extended Papers from Science and Information*, 2014.

J. N. Binongo. Who wrote the 15th book of oz? an application of multivariate analysis to authorship attribution. *Chance*, 16(2):9–17, 2003.

B. Blamey, T. Crick, and G. Oatley. Ru:-) or:-(? character-vs. word-gram feature selection for sentiment classification of osn corpora. In *Research and development in intelligent systems XXIX*, 2012.

L. C. Cagnina and P. Rosso. Classification of deceptive opinions using a low dimensionality representation. In *6TH Workshop on computational approaches to subjectivity, sentiment and social media analysis, EMNLP, páginas 58–66*, 2015.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res*, 7:1–30, 2006.

H. Drucker, D. Wu, and V.N. Vapnik. Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on*, 10(5):1048–1054, 2002.

W. Duan, B. Gu, and A. Whinston. Do online reviews matter? an empirical investigation of panel data. *Decision Support Systems*, 45(4):1007–1016, 2008.

C. Elkan. Evaluating classifiers. In *University of San Diego, California*, 2012.

S. Feng, R. Banerjee, and Y. Choi. Syntactic stylometry for deception detection. In *Association for Computational Linguistics*, 2012a.

S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In *Proceedings of the 2012 international AAAI conference on WebBlogs and Social Media*, 2012b.

V. M. Feng and G. Hirst. Detecting deceptive opinions with profile compatibility. In *Proceedings of the 6th International Joint Conference on Natural Language Processing, páginas 338–346*, 2013.

E. Ferretti, D. Hernández Fusilier, R. Guzmán-Cabrera, M. Montes y Gómez, M. Errecalde, and P. Rosso. On the use of pu learning for quality flaw prediction in wikipedia. In *CLEF 2012 Evaluation labs and workshop, On line working notes papers, Roma, Italia, Septiembre 17-20*, 2012.

E. Fitzpatrick, J. Bachenko, and T Fornaciari. *Automatic Detection of Verbal Deception*. Morgan & Claypool, Publishers, 2015.

G. Forman and I. Cohen. Learning from little: Comparison of classifiers given little training. In *Knowledge Discovery in Databases. Lecture Notes in Computer Science, PKDD, páginas 161–172*, 2004.

P. Graham. *Hackers & painters: big ideas from the computer age*. O'Reilly Media, Inc., 2004.

Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trust rank. In *Proceedings of the Thirtieth international conference on Very large data bases, páginas 576–587*, 2004.

M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1): 10–18, 2009.

D. Hernández, R. Guzmán, M. Montes y Gómez, and P. Rosso. Using pu-learning to detect deceptive opinion spam. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis for Computational Linguistics, NAACL, páginas 38–45, Atlanta, USA, Junio 14*, 2013.

D. Hernández, M. Montes y Gómez, P. Rosso., and R. Guzmán. Detecting positive and negative deceptive opinions using pu-learning. *Information Processing & Management*, 51:433–443, 2015a.

D. Hernández, M. Montes y Gómez, P. Rosso, and R. Guzmán. Detection of opinion spam with character n-grams. In *Computational Linguistics and Intelligent Text Processing. Lecture Notes in Computer Science, Volumen 9042, Parte II, páginas 285–294*, 2015b.

P. Jaccard. *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

N. Japkowicz and S. Shaju. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

N. Jindal and B. Liu. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining, ACM, páginas 219–230*, 2008.

N. Jindal, B. Liu., and E. P. Lim. Finding unusual review patterns using unexpected rules. In *CIKM, ACM, páginas 219–230*, 2010.

I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos. Word versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(6):1047–1067, 2007.

C. Lee and G. G. Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing & Management*, 42(1):155–165, 2006.

D. Lewis. Naïve bayes at forty: The independence assumption in information retrieva. In *Springer Berlin Heidelberg*, 1998.

J. Li, M. Ott, C. Cardie, and E. Hovy. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL*, 2013.

E.P. Lim, V.A. Nguyen, N. Jindal, B. Liu, and H.W. Lauw. Detecting product review spammers using rating behaviors. In *In CIKM, ACM, páginas 939–948*, 2010.

B. Liu. *A Statistical Language Modeling Approach to Online Deception Detection.* Morgan & Claypool Publishers, 2012.

B. Liu, Y. Dai, X. L. Li, W. S. Lee, and Y. Philip. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002), páginas 387–394*, 2002.

B. Liu, Y. Dai, X.L. Li, W.S. Lee, and Philip Y. Building text classifiers using positive and unlabeled examples. In *ICDM-03, páginas 19–22*, 2003.

L. M. Manevitz and M. Yousef. One-class svms for document classification. In *J. Mach. Learn. Res, JMLR.org, páginas 139–154*, 2002.

R. Mihalcea and C. Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. Association for Computational Linguistics, páginas 309–312*, 2009.

A. Mukherjee, B. Liu, J. Wang, N. Glance, and N. Jindal. Detecting group review spam. In *Proceedings of the 20th international conference companion on World wide web, ACM, páginas 93–94*, 2011.

A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Transactions on Management Information Systems (TMIS), ACM, páginas 83–92.*, 2006.

M. Ott, Y. Choi, C.Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, ACL, páginas 309–319*, 2011.

M. Ott, C. Cardie, and J. T. Hancock. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics, ACL*, 2013.

Y. K. Raymond, S. Lau, R. Liao, K. Chi-Wai, X. Kaiquan, X. Yunqing, and L. Yuefeng. Text mining and probabilistic modeling for online review spam detection. *ACM Transactions on Management Information Systems*, 2(4):1–30, 2011a.

Y. K. Raymond, S. Y. Lau, K. R. Chi-Wai Liao, X. Kaiquan, X. Yunqing, and L. Yuefeng. Text mining and probabilistic modeling for online review spam detection. In *Proceedings of the international conference on Web search and web data mining, Volume 2, Issue 4, Article 25*, 2011b.

Y. Ren, D. Ji, and H. Zhang. Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2014.

A. Reyes and P. Rosso. Making objective decisions from subjective data: Detecting irony in customers reviews. In *Journal on Decision Support Systems, vol. 53, issue 4 (Special Issue on Computational Approaches to Subjectivity and Sentiment Analysis) DOI: 10.1016/j.dss.2012.05.027, páginas 754–760*, 2012.

X. Sihong, W. Guan, L. Shuyang, and S. Yu. Philip. Review spam detection via time series pattern discovery. In *Proceedings of the 21st international conference companion on World Wide Web, ACM, páginas 635–636*, 2012.

J. Song, S. Lee, and J. Kim. Spam filtering in twitter using sender-receiver relationship. In *Recent Advances in Intrusion Detection*, 2011.

E. Stamatatos. On the robustness of authorship attribution based on character n-gram features. *Journal of Law & Policy*, 21(2):421–439, 2012.

D. M. J. Tax. *One-class clasification*. Technische Universiteit Delft, 2001.

D. M. J. Tax and R. Duin. Combining one-class classifiers. In *Multiple Classifier Systems*, 2001.

F. Wilcoxon. Individual comparisons by ranking methods. In *Biometrics bulletin*, 1945.

G. Wu, D. Greene, and P. Cunningham. Merging multiple criteria to identify suspicious reviews. In *RecSys'10, ACM, páginas 241–244*, 2010.

B. Zhang and W. Zuo. Reliable negative extracting based on knn for learning from positive and unlabeled examples. *Journal of Computers*, 4:94–101, 2009.

L. Zhou, Y. Sh, and D. Zhang. Merging multiple criteria to identify suspicious reviews. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1077–1081, 2008.