Detección automática de texto engañoso mediante un modelo de espacio semántico continuo

Edición bilingüe



Ángel Hernández Castañeda
Hiram Calvo
René Arnulfo García Hernández

Luis Cejudo Espinosa
Traductor

PUBLICACIONES
UAEM

Detección automática de texto engañoso mediante un modelo de espacio semántico continuo

Edición bilingüe

Universidad Autónoma del Estado de México

Dr. en Ed. Alfredo Barrera Baca
Rector

Dr. en C. I. Amb. Carlos Eduardo Barrera Díaz
Secretario de Investigación y Estudios Avanzados

Dr. en Ing. Martín Carlos Vera Estrada
Coordinador de la Unidad Académica Profesional Tianguistenco

Mtra. en Admón. Susana García Hernández
Directora de Difusión y Promoción de la Investigación y los Estudios Avanzados

Detección automática de texto engañoso mediante un modelo de espacio semántico continuo

Ángel Hernández Castañeda
Hiram Calvo
René Arnulfo García Hernández

Luis Cejudo Espinosa
Traductor

Toluca, 2020

Detección automática de texto engañoso
mediante un modelo de espacio semántico continuo

Ángel Hernández Castañeda
Hiram Calvo
René Arnulfo García Hernández

Primera edición: noviembre de 2020
ISBN 978-607-633-222-1 (electrónico)
ISBN 978-607-633-221-4 (impreso)

D. R. © Universidad Autónoma del Estado de México
Instituto Literario núm. 100 Ote.
C.P. 50000, Toluca, Estado de México
www.uaemex.mx

Portada: Juan Manuel García Guerrero
Traducción: Luis Cejudo Espinosa

El presente libro cuenta con la revisión y aprobación de dos pares doble ciego, adscritos al Sistema Nacional de Investigadores, externos a la Universidad Autónoma del Estado de México. El arbitraje estuvo a cargo de la Secretaría de Investigación y Estudios Avanzados, según consta en el expediente 234/2019.

Esta edición y sus características son propiedad de la Universidad Autónoma del Estado de México.

El contenido de esta publicación es responsabilidad de los autores.

Los recursos del Profexce (Programa de Fortalecimiento a la Excelencia Educativa) son de carácter público y queda prohibido su uso con fines partidistas o de promoción personal.

Índice

Introducción **14**

1. La detección de texto engañoso **16**

1.1. ¿Por qué es importante detectar el engaño en los textos? 20
1.2. Identificar texto engañoso es todo un reto 22

2. Conceptos básicos **24**
2.1. Procesamiento de texto 24
2.2. Tokenización 27
2.3. Análisis sintáctico 28
2.3.1. Gramáticas libres del contexto 30
2.3.2. Árboles sintácticos 31
2.3.3. Flexibilidad 32
2.3.4. Desambiguación 35
2.4. Análisis semántico 35
2.4.1. Teorías y enfoques de la representación semántica 38
2.4.2. Enfoques lógicos 38
2.4.3. Teoría de representación de discurso 39
2.4.3.1. Lexicón generativo de Pustejovsky 40
2.4.3.2. Metalenguaje semántico natural 41
2.4.3.3. Semántica orientada a objetos 42
2.4.3.4. Hechos relacionales y ontologías 42

3. Métodos para la identificación del engaño **44**

3.1. Antecedentes 44
3.2. Detección de engaño en tópicos controversiales 47
3.3. Detección de engaño en opiniones sobre hoteles 50
3.4. Detección de engaño en declaraciones 55
3.5. Detección de engaño en mezcla de dominios 57
3.6. Detección de engaño en opiniones sobre libros 61

4. Un nuevo enfoque para la detección de engaño **66**

4.1. Fuentes de generación de características 66
4.1.1. Matriz término-documento (MTD) 67
4.1.2. N-gramas sintácticos 68
4.1.3. Linguistic Inquiry and Word Count 71

4.1.4. Modelo de espacio semántico continuo-Latent Dirichlet Allocation (LDA) 72
4.2. Conjuntos de datos 75
4.2.1. Conjunto de datos DeRev 75
4.2.2. Conjunto de datos OpSpam 77
4.2.3. Conjunto de datos de tópicos controversiales 78
4.2.4. Análisis de los conjuntos de datos 79
4.3. Construcción de vectores de características 81
4.4. Identificación del número de tópicos óptimo 84

5. Evaluación del enfoque propuesto 86

5.1. Detección de engaño 86
5.1.1. Medidas de validación 87
5.1.1.1. Medida F 88
5.1.1.2. Área bajo la curva ROC 89
5.1.2. Clasificación de un dominio específico 90
5.1.3. Clasificación de dominios mezclados 94
5.1.4. Clasificación de dominio cruzado 96
5.1.5. Palabras y tópicos predominantes 99
5.1.6. Comparación de resultados y significancia estadística 103
5.2. Impacto de la polaridad en los textos engañosos 107
5.2.1. Clasificación de polaridad para mejorar la detección de engaño 111
5.3. Resolución de una tarea diferente 114
5.3.1. Identificación de autoría 115
5.3.2. Enfoques para la verificación de autoría 116
5.3.3. Conjuntos de datos 119
5.3.3.1. Conjunto de datos PAN 2015 120
5.3.3.2. Conjunto de datos PAN 2014 121
5.3.4. Método propuesto 121
5.3.4.1. Método para la verificación de autoría 122
5.3.5. Resultados 125
5.3.6. Resultados de la clasificación de PAN 2015 126
5.3.7. Resultados de la clasificación de PAN 2014 133

Conclusión 136
Referencias 138

Índice de figuras

II.1. Ejemplo de árbol sintáctico 33

IV.1. Ejemplo de construcción de vectores mediante un modelo de espacio de palabras 70

IV.2. Proceso de generación de bigramas sintácticos dada una oración 71

IV.3. Ejemplo de algunos grupos de palabras, en LIWC, con su respectiva etiqueta. Es mostrado un grupo de sólo 7 palabras por etiqueta 72

IV.4. Ejemplo de generación de tópicos, de textos del tema *pena de muerte*, mediante LDA 74

IV.5. Ejemplo de documento procesado por LDA. Se muestra la distribución de tópicos 74

V.1. Precisión y exhaustividad 89

V.2. Curva de aprendizaje de los diferentes corpora 93

V.3. Ejemplo de resta entre el vector de documento conocido y el vector de documento desconocido 125

V.4. Diferencias de tópicos entre documentos escritos por el mismo autor o por un autor diferente (subconjunto en inglés) 129

V.5. Diferencias de tópicos entre documentos escritos por el mismo autor o por un autor diferente (subconjunto en español) 129

Índice de tablas

III.1. Clases de LIWC dominantes del texto engañoso y veraz 51

III.2. Exactitud obtenida por tópico usando una validación cruzada de 10 pliegues 51

III.3. Exactitud, precisión (P), exahustividad (R), y medida F (F) de la identificación de texto engañoso realizada por humanos 54

III.4. Resultados de la aplicación de diferentes fuentes de generación de características y aplicación de dos clasificadores 54

III.5. Resultados de la clasificación de corpus DeCour. Se muestran: precisión (P), exhaustividad (R) y medida F (F), asimismo el promedio de medidas F 57

III.6. Experimentos realizados sobre los corpora de cada país, por separado 59

III.7. Clasificación de cultura cruzada con LIWC y unigramas como fuente de características 60

III.8. Clases de LIWC predominantes en el texto engañoso y en el texto veraz 62

III.9. Clasificación del conjunto DeRev que muestra el rendimiento de los métodos de etiquetado 64

IV.1. Comparación entre bigramas sintácticos y convencionales en la tarea de identificación de autoría 69

IV.2. Comparación entre trigramas sintácticos y convencionales en la tarea de identificación de autoría 70

IV.3. Cantidad de tipos y *tokens* en los corpora. PTD = Promedio de *Tokens* por Documento 80

IV.4. Promedio de tipos compartidos en los diferentes corpora 80

IV.5. Comparación de exactitud (%) con respecto a los valores binarios y la selección de atributos (SeAt) 84

IV.6. Medida F obtenida con diferentes valores del número de tópicos 84

V.1. Clasificación del corpus OpSpam mediante el uso de diferentes fuentes de generación de características 94

V.2. Número de características relevantes obtenidas con la combinación de LDA y MTD 95

V.3. Clasificación del corpus DeRev mediante el uso de diferentes fuentes degeneración de características 95

V.4. Clasificación del tópico *aborto* mediante el uso de diferentes fuentes de generación de características 96

V.5. Clasificación del tópico *mejor amigo* mediante el uso de diferentes fuentes de generación de características 97

V.6. Clasificación del tópico *pena de muerte* mediante el uso de diferentes fuentes de generación de características 98

V.7. Exactitud, precisión (P), exhaustividad (R) y medida F (F) obtenidas con los corpora mezclados mediante SVM 98

V.8. Exactitud, precisión (P), exhaustividad (R) y medida F (F) obtenidas con los corpora mezclados mediante NB 99

V.9. Medida F obtenida en la clasificación de dominio cruzado. Se muestran resultados para los clasificadores NB y SVM 100

V.10. Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos OpSpam 102

V.11. Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos *mejores amigos* 102

V.12. Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz en el conjunto de datos *aborto* 103

V.13. Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos *pena de muerte* 104

V.14. Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos DeRev 105

V.15. Comparación de nuestros resultados con otros estudios de los mismos corpora 106

V.16. Significancia estadística 107

V.17. Resultados de la clasificación excluyendo palabras de polaridad 110

V.18. Clasificación del corpus OpSpam, etiquetado por los autores (OAP) y el etiquetado realizado en este estudio (PTW), basado en la polaridad del texto 111

V.19. Clasificación de los conjuntos de datos agregando característica de polaridad 111

V.20. Comparación de la medida f obtenida antes (F-Ant) y después (Avg. F) de la clasificación de polaridad. Las medidas precision (P), Recall (R), medida F (F), y el número de opiniones por clase (# docs.) son también mostrados 112

V.21. Detalles del conjunto de datos para la tarea PAN 2015 de identificación del autor 119

V.22. Detalles del conjunto de datos para la tarea PAN 2014 de identificación del autor 119

V.23. Rendimiento obtenido en el idioma inglés, ajustando el número de tópicos 126

V.24. Rendimiento obtenido en el idioma español, ajustando el número de tópicos 127

V.25. Rendimiento obtenido en el idioma holandés, ajustando el número de tópicos 127

V.26. Rendimiento obtenido en el idioma griego, ajustando el número de tópicos 128

V.27. Resultados de cada clasificación de subconjunto. Exactitud, precisión (P), exahustividad (R) y Medida F (F) 131

V.28. Comparación de resultados con otros autores. FS=C@1*AUC 132

V.29. Resultados detallados de clasificación en géneros donde se alcanzó la línea de base 132

V.30. Resultados detallados de clasificación en géneros donde no se alcanzó la línea de base 133

V.31. Comparativa de desempeño de este trabajo y líneas de base PAN 2014 133

V.32. Comparativo de desempeño de este estudio y los participantes de PAN 2014 134

Introducción

La Inteligencia Artificial (IA) es un área de estudio que busca replicar los procesos cognitivos de los cuales los humanos están dotados para poder incorporarlos a los ordenadores. Una línea de estudio particular de dicha área es el Procesamiento del Lenguaje Natural (PLN), el cual tiene como objetivo facilitar la comunicación entre las computadoras y el lenguaje humano.

Derivado de una tesis doctoral, este libro presenta un nuevo método para identificar el engaño en los textos mediante técnicas del PLN y algoritmos de aprendizaje automático, está dirigido a estudiantes e investigadores que se interesan por la IA y el PLN, específicamente en la generación de modelos para la detección de texto engañoso.

El proceso general que sigue el enfoque propuesto en este texto es el siguiente: mediante herramientas del PLN los textos se convierten a un espacio vectorial, es decir, se generan vectores de características; después se aplicaron diversos métodos de selección de atributos para discriminar aquellas características poco relevantes; por último, los vectores resultantes se enviaron a un algoritmo de clasificación supervisada que identificó si son engañosos o veraces.

Cabe destacar que los objetivos de esta investigación fueron experimentar con modelos basados en tópicos, particularmente LDA

(Latent Dirichlet Allocation) con el fin de detectar el engaño en los texos de estudio; asimismo, combinar características generadas mediante diferentes enfoques con el fin de mejorar la eficiencia de la detección de engaño; no sin dejar a un lado evaluar tanto el desempeño de enfoques seleccionados (por separado y combinados) en diferentes casos de estudio, como el rendimiento de los métodos y combinación de éstos en la clasificación de dominios específico, mezclado, cruzado y el impacto de la polaridad en la clasidicación de textos engañosos; así como aplicar el enfoque propuesto en otras áreas del PLN.

Por ello, el presente libro consta de cinco capítulos: en el primero se da una breve introducción sobre el texto engañoso y los enfoques experimentales manejados en esta investigación; en el segundo, se presenta el fundamento teórico que procura las bases para comprender las herramientas de generación de características aplicadas. El tercero muestra un resumen de diferentes estudios que proponen enfoques, mediante aprendizaje automático, para la identificación de texto engañoso. En el capítulo cuatro, se estudia el enfoque propuesto en el presente texto, además de los conjuntos de datos analizados y clasificados. El capítulo cinco, y último, detalla los resultados obtenidos en diferentes casos analizados.

Capítulo 1

La detección de texto engañoso

El estudio del engaño se ha perfeccionado con el tiempo, desde el análisis del rostro, por ejemplo, mediante el movimiento de los ojos y las expresiones faciales; hasta el monitoreo de las respuestas fisiológicas del ser humano mediante un polígrafo. Este último enfoque ha logrado resultados óptimos (con una exactitud por encima del 95 %) en la detección del engaño. Los principios del funcionamiento de estos detectores de engaño se basan principalmente en la siguiente hipótesis: mentir produce cierto estado emocional y para cada estado emocional existe una reacción fisiológica identificable por el detector de engaño revisando variables, tales como la presión arterial, el ritmo cardíaco y la frecuencia respiratoria.

De manera similar, investigadores del PLN han tratado de identificar el engaño en el texto porque cuando la gente trata de engañar mediante la escritura usa ciertas palabras que en un texto real no usaría; es decir, el engaño propicia un estado emocional que cambia nuestra forma de escribir. Por ejemplo, cuando al estar realizando una tarea en especial pasamos de nuestro estado normal a una situación muy estresante es

posible que no recordemos o que omitamos ciertos detalles de la tarea en marcha debido al momento de estrés en el que estamos.

Escribir un texto engañoso a una persona o conjunto de personas puede generar, de forma consciente o inconsciente, un cambio en el comportamiento; o mejor dicho en la forma habitual en que realizamos una acción. Este cambio se genera debido a que el cerebro siempre está preparado para decir la verdad; sin embargo, para mentir debe incrementar el rendimiento de los procesos cognitivos.

Al respecto, Ekman (2009: 26) menciona: "Un mentiroso puede elegir no mentir. Engañar a la víctima es un acto deliberado; el mentiroso pretende desinformar a la víctima. El mentiroso puede estar o no estar justificado, bajo su opinión o bajo la opinión de la comunidad. El mentiroso puede ser buena o mala persona, agradable o desagradable. Sin embargo, la persona que ha mentido pudo elegir engañar o ser veraz, y sabe la diferencia entre ambas".

Se puede dar un debate filosófico de lo que puede significar el concepto *mentir* puesto que no existe una definición universal; sin embargo, para fines de este estudio, se considera que el engaño se presenta cuando el mentiroso pretende transferir a la víctima información total o parcialmente falsa, es decir, hacer creer a la víctima sobre un hecho que no sucedió o que aconteció de manera diferente. Asimismo, se buscó mejorar la detección de engaño en opiniones de distintos dominios.

Hoy en día, los usuarios realizan una variedad de actividades en línea tales como comprar y vender artículos, difundir ideas mediante blogs e intercambiar información en general. Dicha información no siempre es confiable: algunas personas utilizan internet para transmitir información con el propósito de manipular y engañar a otros usuarios. Así, cuando un

usuario quiere comprar un artículo en línea, la principal manera de saber si el producto es bueno, o no, es leer la sección de opiniones sobre éste en la página web del vendedor. Tales opiniones han demostrado tener un gran impacto en la decisión final de adquirir, o no, el artículo. Debido a esto, algunos vendedores contratan a personas, algunas veces expertas, para escribir opiniones positivas con el fin de aumentar las ventas de un producto, incluso si esas personas no tienen una idea real sobre la calidad o cualidad sobre éste. En otros casos, las opiniones engañosas pretenden desacreditar los productos ofrecidos por los competidores.

Además de los textos engañosos escritos para manipular las decisiones de compra de los usuarios, también existen textos engañosos que pretenden cambiar la opinión o el punto de vista de las personas sobre un tema determinado, como un candidato político o un tema de debate público. Todo esto hace muy importante el estudio de la detección de textos engañosos. La tarea, básicamente, puede definirse como la identificación de aquellas opiniones escritas en las que el autor pretende transmitir información en la que no cree (Keila y Skillicorn, 2005).

Los estudios sobre textos engañosos han demostrado empíricamente que la comunicación veraz es cualitativamente diferente de la comunicación engañosa (Ekman, 1989; Twitchell *et al.*, 2004). Debido a esto, se han emprendido diferentes proyectos con el objetivo de identificar los textos engañosos con la mayor exactitud posible. Para ello, se han creado diversos conjuntos de datos que consisten en textos etiquetados como veraces o engañosos y permiten medir la eficiencia de los enfoques de detección propuestos.

En el enfoque de aprendizaje automático, una parte de los textos del conjunto de datos se utiliza como un conjunto de entrenamiento para

un clasificador y el resto como un conjunto de prueba. Por lo tanto, una comparación directa entre diferentes clasificadores y métodos de selección de características es posible mediante la aplicación de éstos sobre el mismo conjunto de datos.

Para probar la eficiencia en la detección de texto engañoso, se procesaron tres conjuntos de datos sobre diferentes tópicos, específicamente: OpSpam, que consiste en opiniones sobre hoteles; DeRev, que consiste en opiniones sobre libros comprados en Amazon y el conjunto de datos de tópicos controversiales, compuesto de opiniones sobre tres temas (*aborto*, *pena de muerte* y *mejor amigo*).

Además se analizaron diferentes casos de estudio. En primer lugar, se evaluó cada conjunto de datos por separado (sección 5.1.2) en un esquema donde los textos se refieren a un mismo tema o dominio; en segundo, se realizaron experimentos mezclando todos los conjuntos de datos (sección 4.2) para verificar si las características obtenidas de una mezcla de dominios (sección 5.1.3) mejoraban los resultados o, por el contrario, los empeoraban; finalmente, se exploró si un conjunto de características puede ser suficientemente general para ser usado en la clasificación de un conjunto de datos que trate de un tema diferente al del conjunto de datos utilizado para su formación, lo que permitiría crear detectores de texto engañoso de propósito general independientes del dominio (sección 5.1.4).

Para poder hacer uso de un clasificador supervisado es necesario mapear los textos a vectores numéricos, para ello, se generaron características mediante el uso de diversos métodos, tales como Latent Dirichlet Allocation (LDA), Linguistic Inquiry and Word Count (LIWC),

y un Modelo de Espacio de Palabras (MEP), así como combinaciones de características generadas por los diferentes métodos.

Con base en los conjuntos de datos disponibles, se investigó qué método es mejor en un único dominio, donde tanto el conjunto de entrenamiento y de prueba son del mismo tema; en un entorno de dominio mixto, donde tanto el conjunto de entrenamiento y de prueba consisten en una mezcla de tópicos y en un entorno de dominio cruzado, donde los conjuntos de entrenamiento y prueba están formados de diferentes tópicos (sin intersección entre prueba y entrenamiento).

Con la ejecución de estos experimentos, se pudo evaluar la posibilidad de utilizar conjuntos de datos existentes para detectar textos engañosos sobre un tema para el que no hay un conjunto de datos disponible, es decir, la posibilidad de desarrollar un detector de texto engañoso independiente del dominio.

1.1. ¿Por qué es importante detectar el engaño en los textos?

En la actualidad, la venta de productos y servicios está migrando a la web. A diferencia de una venta en donde el producto está físicamente presente para poder revisarlo, en una venta por internet sólo se cuenta con un conjunto de imágenes para revisar lo que se está comprando.

Los vendedores consideraron que las imágenes no bastaban para que el comprador tuviera la seguridad de que el producto cumpliría sus expectativas. Por lo tanto, se añadió una sección en donde los compradores pueden agregar sus comentarios después de tener y probar el producto. Éstos servirían para que los futuros compradores pongan en tela de juicio su compra, es decir, si las opiniones son positivas en su

mayoría, entonces el comprador sabrá que el producto es bueno, de otra forma, si existen demasiadas opiniones negativas, el comprador tendrá los argumentos necesarios para tomar la decisión de comprar, o no, el producto en cuestión.

Esto fue una buena idea hasta que los compradores lo vieron como una forma de sacar ventaja y aumentar sus ventas. Como resultado, diferentes páginas en internet ofrecen productos y buscan a personas para escribir opiniones engañosas acerca de los productos, dichas opiniones dirán que el producto es muy bueno cuando en realidad la persona que lo escribió tendrá como fin ayudar al vendedor a generar más ventas.

Las opiniones engañosas no son necesariamente un argumento positivo hacia el producto. Puede existir el caso en que la competencia quiera desprestigiar a determinado vendedor, por lo cual tendrá que agregar un comentario que no favorezca al producto en cuestión. Esta opinión será por tanto negativa y también engañosa.

En un artículo[1] publicado en la revista *The Economist* en 2015 se concluyó que más de 1 000 usuarios en Amazon[2] se dedican a escribir opiniones engañosas. En un caso particular, de acuerdo con este artículo, un restaurante con una estrella más en Yelp (un sitio popular de opiniones) aumentará sus ingresos de 5 % a 9 %. Además de Amazon, en muchos otros sitios web existen este tipo de opiniones debido a que la motivación es clara, un aumento en las ventas.

La propagación de este tipo de opiniones engañosas se debe principalmente a que los humanos tenemos problemas para detectar el engaño. Esto se demuestra en la investigación de Pérez Rosas *et al.*

[1] http://www.economist.com/news/business/21676835-evolving-fight-against-sham-reviews-five-star-fakes

[2] www.amazon.com

(2015), donde se verifica el rendimiento de tres jueces humanos en la detección de engaño en el texto, audio y video. Asimismo, se muestra que el rendimiento oscila entre 54.2-65.3 %, 58.5-70.3 %, 63.0-71.0 % para texto, audio y video, respectivamente. Se puede notar que en el caso del texto es más difícil poder diferenciar el engaño de la verdad. Esto puede deberse a que no podemos observar al sujeto que está mintiendo, por lo tanto, no se puede obtener cierta información; por ejemplo, si el sujeto está nervioso, con voz agitada, está sudando o alguna señal que pueda ayudar a mejorar el rendimiento en la identificación del engaño.

Dado lo anterior, es necesaria la investigación de nuevos enfoques que permitan detectar, con mayor eficiencia, el engaño en los textos. De esta manera, se proveerán mecanismos que ayuden a los compradores de distintas páginas de internet para que puedan realizar compras más seguras y confiables.

1.2. Identificar texto engañoso es todo un reto

La detección de engaño en un texto es un reto aún mayor que detectar engaño en una persona debido a que sólo se cuenta con las palabras incluidas en el documento o colección de documentos. Es decir, no tenemos a la persona para monitorear sus reacciones fisiológicas, sus expresiones faciales o incluso su comportamiento.

En distintos estudios se han tratado de buscar palabras que identifiquen al texto engañoso, sin embargo, se pueden formular las siguientes preguntas: ¿estas palabras se mantienen para distintos tipos de engaño?, ¿las palabras que identifican al texto engañoso pueden cambiar si el tema al que se refiere el texto cambia también?, ¿cuál es el método de

generación de características que genera un modelo que represente de la mejor manera al texto engañoso?, ¿la combinación de distintos métodos puede generar un mejor resultado en la detección de texto engañoso?

Es difícil saber si las personas tienen un comportamiento predeterminado al realizar cierta acción tal como mentir. Sin embargo, con base en distintos trabajos que se realizaron previamente (sección 3), se ha logrado identificar que los mentirosos siguen un patrón de uso de palabras; por ejemplo, al momento de mentir pueden ser poco específicos, usar en mayor o menor cantidad ciertas palabras.

Estas investigaciones prueban que, mediante algoritmos de reconocimiento de patrones es posible distinguir un texto engañoso de un texto veraz; aunque existen otras limitantes que todavía tienen que ser analizadas, por ejemplo, la falta de conjuntos de datos (o los conjuntos generalmente pequeños) que no permiten que los enfoques propuestos sean probados en un entorno más general.

Capítulo 2

Conceptos básicos

2.1. Procesamiento de texto

El lenguaje natural contiene ambigüedades inherentes, y el lenguaje escrito genera aún más ambigüedad (Indurkhya *et al.*, 2010).

El procesamiento de texto es la tarea de convertir una cantidad de archivos de texto en una secuencia bien definida de unidades lingüísticas significativas, es parte esencial de cualquier sistema de PLN debido a que los caracteres, palabras y sentencias identificadas en esta fase son las unidades fundamentales que pasan a las siguientes fases de procesamiento. Estas fases pueden ser de análisis y de etiquetado, por ejemplo, análisis morfológico y etiquetado de categorías gramaticales, por medio de aplicaciones como recuperación de información y sistemas de traducción automática.

El procesamiento de texto puede dividirse en dos etapas: en primer lugar, el proceso de convertir un conjunto de archivos digitales en un conjunto de documentos de texto bien definidos. Esto implica, también, determinar los algoritmos para analizar el lenguaje específico de los documentos; la identificación del lenguaje determina, entonces,

el lenguaje natural del cual están formados los documentos. Este paso está muy relacionado con la codificación de caracteres (aunque esto no lo determina únicamente).

En la segunda, está la segmentación de texto que es el proceso de convertir un conjunto de datos de texto bien definido en componentes como palabras y sentencias. La segmentación consiste en partir o cortar las secuencias de caracteres en el texto mediante la localización de los límites de las palabras, es decir, los puntos donde comienza y termina una palabra. Para los propósitos de la lingüística computacional, a las palabras identificadas, con la segmentación de texto, se les da el nombre de *tokens*; además, este proceso también es conocido como tokenización.

Por una parte, la normalización del texto es un proceso relacionado, el cual incluye todas las mezclas de la escritura de un token en una forma canónica normalizada; por ejemplo, las diferentes formas del token: *Señor*, *Sr.*, *señor*, *SEÑOR*. Por otra parte, también se puede incluir el proceso de segmentación de sentencias que identifica dónde comienza y termina una sentencia.

El tipo de sistema de escritura usado por un lenguaje es el factor más importante para determinar el mejor enfoque de preprocesamiento. Los sistemas de escritura pueden ser logográficos, donde un extenso número de símbolos individuales representan palabras; silábicos, donde los símbolos individuales representan sílabas; alfabéticos, donde los símbolos individuales representan sonidos. En la práctica, no hay sistemas modernos de escritura que usen símbolos de un solo tipo, por lo tanto, ningún sistema de lenguaje natural de escritura puede ser clasificado como puramente logográfico, silábico o alfabético. El inglés, por ejemplo, está basado en el alfabeto latino y utiliza símbolos

logográficos incluyendo los números arábigos, símbolos de moneda, entre otros. Dicho alfabeto, sin embargo, es predominantemente alfabético, y muchos de los otros sistemas de escritura están compuestos por símbolos que son mayormente de un tipo.

La tokenización está inevitablemente relacionada con la subyacente codificación de caracteres del texto que se va a procesar, y la identificación de caracteres es un primer paso esencial. Mientras la cabecera de un documento digital puede contener la información correspondiente con su codificación de caracteres, esta información no está siempre presente o incluso disponible, en este caso, la codificación debe ser determinada automáticamente.

Además de la variedad de tipos de símbolos (logográficos, silábicos o alfabéticos) usados en los sistemas de escritura, existen un conjunto de convenciones ortográficas en los lenguajes escritos para denotar los límites entre unidades lingüísticas tal como sílabas, palabras o sentencias. El inglés y maneja los espacios en blanco entre palabras y signos de puntuación como límite de sentencia; sin embargo, incluso esto no es suficiente para segmentar completamente el texto solventando las ambigüedades que se presentan.

Muchos de los algoritmos de segmentación de lenguajes naturales existentes son para lenguajes específicos y dependientes de corpus, desarrollados para manejar las ambigüedades predecibles en un texto bien formado. Dependiendo del origen y del propósito del texto, el uso de letras mayúsculas y las reglas de puntuación pueden ser respetadas como en los documentos científicos, medianamente respetadas como en algunas páginas web informativas y prácticamente descuidadas como en las redes sociales.

Aunque la segmentación de palabras y sentencias es necesaria, en realidad, no existe una definición absoluta sobre lo que constituye cada una de éstas. Ambas tienen distinciones relativamente arbitrarias que cambian fuertemente en los diferentes lenguajes escritos. Sin embargo, para propósitos de la computación lingüística, se requiere definir exactamente qué se necesita para un procesamiento posterior; en muchos casos, el lenguaje y la aplicación determinan las convenciones necesarias.

2.2. Tokenización

La tokenización, en la actualidad, está bien establecida y entendida para los lenguajes artificiales, tal como los lenguajes de programación. Sin embargo, tales lenguajes artificiales pueden ser definidos para eliminar las ambigüedades léxicas y estructurales; en el caso del lenguaje natural no se puede contar con la misma definición; en este tipo de lenguajes una palabra puede servir para diferentes propósitos y la sintaxis no está estrictamente definida.

Hay lenguajes delimitados por espacios, por una parte, tal como muchos lenguajes europeos en los cuales los límites de palabra son indicados por la inserción de un espacio en blanco. Por otra parte, también hay lenguajes no segmentados, como el chino y thai, donde las palabras son escritas consecutivamente sin indicar ningún límite entre las mismas.

Hay tres categorías principales en las cuales las estructuras de las palabras pueden ser colocadas y cada categoría existe en ambos sistemas de escritura: delimitados por espacio y no segmentados. La morfología de las palabras en un lenguaje puede ser aislada, cuando

las palabras no se dividen en pequeñas unidades; aglutinante, cuando las palabras se dividen en pequeñas unidades, llamadas morfemas, con límites claros entre ellos; o flexional, cuando los límites entre morfemas no son claros y los morfemas componentes pueden expresar más de un sentido gramatical.

Aun contando con un corpus de sentencias bien formado, existen varios tópicos por resolver en la tokenización. Mucha de la ambigüedad, al tokenizar, se presenta por el uso de signos de puntuación debido a que el mismo signo de puntuación puede tener diferentes funciones en una sola sentencia.

Una idea inicial en la tokenización de un lenguaje delimitado por espacios se podría considerar como un token separado por cualquier secuencia de caracteres precedidos y seguidos de un espacio. Esto tokeniza exitosamente palabras que son una secuencia de signos alfabéticos, pero no toma en cuenta caracteres de puntuación. En muchos casos, los caracteres tal como comas, punto y coma y puntos deben ser tratados como *tokens* separados. La tarea de tokenización en los lenguajes no segmentados necesita un enfoque más elaborado que un simple análisis léxico.

2.3. Análisis sintáctico

Es conveniente considerar los aspectos en los cuales difieren los analizadores sintácticos aplicados a lenguajes naturales y aquellos adaptados a lenguajes de programación. Una de esas diferencias se refiere al poder de los formalismos de la gramática usada, es decir, la capacidad generativa de la gramática. Los lenguajes de programación son

usualmente diseñados para permitir codificar mediante una gramática no ambigua. Para lograrlo se restringen cuidadosamente las subclases de la Gramática Libre del Contexto (CFG), por sus siglas en inglés: Context-Free Grammar). Por el contrario, el lenguaje natural requiere instrumentos más poderosos. Uno de los casos más destacados del poder expresivo han sido las dependencias de larga distancia, por ejemplo, en el inglés, las palabras interrogativas.

Otra diferencia tiene que ver con la estructura ambigua del lenguaje natural. En el recorrido de la sentencia, habrá típicamente reglas gramaticales que podrían aplicar. Un ejemplo clásico es el siguiente: "Pon el cuadro en la caja sobre la mesa". Si asumimos que "pon" subcategoriza a dos objetos, entonces existen dos posibles análisis: "Pon el cuadro [en la caja sobre la mesa]" y "Pon [el cuadro en la caja] sobre la mesa".

El propósito de una gramática general podría ser capturar lo que es posible en cualquier contexto. Mucho del trabajo en el análisis sintáctico trata con las formas en las cuales los potenciales espacios de búsqueda enormes pueden ser tratados eficientemente, y cómo se puede elegir el análisis más apropiado.

La tercera diferencia se basa en el hecho de que los textos en lenguaje natural son inherentemente ruidosos. A diferencia del lenguaje natural, un lenguaje de computadora tiene una sintaxis específica completa, lo cual significa que, por definición, toda cadena de entrada correcta es analizable. En el análisis del lenguaje natural, es difícil distinguir si una falla que produce el resultado del análisis es debido a un error de entrada o a la falta de cobertura de la gramática.

2.3.1. Gramáticas libres del contexto

Desde su introducción por Chomsky (1956), las gramáticas libres del contexto (CFG) han sido el formalismo gramatical más influyente para describir la sintaxis de un lenguaje. Esto debido a que muchos de los formalismos gramaticales se derivan o de alguna manera pueden estar relacionados con las CFG.

La forma estándar de definir una CFG, por una parte, es mediante la tupla $G = (E, N, S, R)$, donde E y N son conjuntos finitos disjuntos con símbolos terminales y no terminales, respectivamente, y el conjunto $V = E \cup N$ contiene los símbolos de la gramática. Por otra parte, R es un conjunto finito de reglas de producción de la forma $A \to \alpha$, donde $A \in N$ es un no terminal y $\alpha \in V$, una secuencia de símbolos.

Se usan letras mayúsculas para identificar a los símbolos no terminales y letras minúsculas para identificar a los símbolos terminales.

Una gramática está en la Forma Normal de Chomsky (CNF, por sus siglas en inglés Chomsky Normal Form) cuando cada regla es terminal unaria de la forma $A \to w$, o una regla binaria de no terminales de la forma $A \to BC$. Siempre es posible transformar una gramática en la CNF tal que acepte el mismo lenguaje. Sin embargo, la transformación puede cambiar radicalmente la estructura de la gramática. Por ejemplo, si la gramática original tiene n reglas, la gramática transformada puede tener, en el peor de los casos, $O(n2)$ reglas.

Se puede atenuar esta forma normal por medio de permitir reglas unitarias de no terminales de la forma $A \to B$. La transformación para esta forma es más simple y la gramática transformada es estructuralmente parecida.

2.3.2. Árboles sintácticos

La forma estándar de representar las estructuras sintácticas de una sentencia gramática es mediante un árbol sintáctico (figura II.1), el cual es una representación de todos los pasos de la derivación de una sentencia desde su nodo raíz. Esto significa que cada nodo interno en el árbol representa una aplicación de una regla gramatical.

En la práctica, las CFG no son ampliamente usadas para implementar gramáticas. Una de las razones es que las CFG no son lo suficientemente expresivas, es decir, no describen todas las peculiaridades del lenguaje natural, pero la razón principal es que son muy difíciles de usar.

En el análisis sintáctico participan diferentes procedimientos. Un reconocedor es una función que determina si una sentencia de entrada se encuentra acorde con la gramática establecida. Asimismo, un analizador sintáctico es un reconocedor que produce análisis estructurales de acuerdo con la gramática. Un analizador robusto produce salidas, tanto como análisis parciales, incluso cuando la entrada no está totalmente cubierta por la gramática.

Podemos pensar que una gramática induce a un espacio de búsqueda que consiste en un conjunto de estados, representando etapas de reescritura de reglas sucesivas, y un conjunto de transiciones entre esos estados. Cuando se analiza una sentencia, el analizador debe reescribir las reglas gramaticales en cierta secuencia. Una secuencia que conecta el estado S, la cadena que consiste de la categoría de inicio de la gramática, y el estado que consiste de la cadena exacta formada por las palabras de entrada es llamada derivación. Cada estado en la secuencia consiste

entonces de un estado en V y es llamada forma sentencial. Si tal secuencia existe, se dice que la sentencia es acorde con la gramática.

Los analizadores sintácticos pueden ser clasificados en diversas dimensiones, de acuerdo con la forma en que se hacen las derivaciones. Una de esas dimensiones se refiere a la regla de invocación. Éstas pueden ser arriba-abajo, donde cada forma sentencial es producida por su predecesor mediante el reemplazo de un símbolo no terminal A por una cadena de símbolos terminales o no terminales $X_1...X_d$, donde $A \rightarrow X_1...X_d$ es una regla gramatical. De forma inversa, en una derivación abajo-arriba, cada forma sentencial es producida al reemplazar $X_1...X_d$ por A, dada la misma regla gramatical, y así sucesivamente aplicando reglas gramaticales en la dirección inversa.

Otra dimensión se refiere al modo en el cual el analizador trata con la ambigüedad, en particular, si el proceso es determinista o no determinista. Para el primer caso, sólo una irrevocable opción puede ser tomada cuando el analizador se enfrenta a una ambigüedad local. Esta elección es típicamente basada en alguna forma de mirar hacia adelante o alguna preferencia sistemática. Una tercera dimensión se refiere a si el analizador procede desde la derecha o desde la izquierda a través de la entrada o en algún otro orden.

2.3.3. Flexibilidad

En un analizador sintáctico es natural esperar que las entradas correspondan con aquellas cadenas que están en el lenguaje formal L(G) generado mediante la gramática G. Sin embargo, un analizador de lenguaje natural siempre podrá estar expuesto a alguna cantidad de entradas que no estarán en L(G). Una fuente de este problema es la

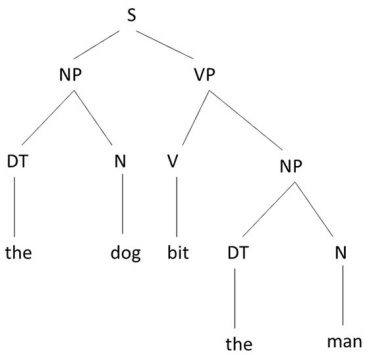


Figura II.1: Ejemplo de árbol sintáctico

subgeneración, la cual es causada por la falta de cobertura de G referente al lenguaje natural G. Otro problema es que la entrada quizá contenga errores, es decir, puede estar mal formada. Pero independientemente de por qué la entrada no está en L(G), es deseable tratar de recuperar tanta información relevante como sea posible, en lugar de no obtener ningún resultado en absoluto. En cierto sentido, el análisis robusto significa que desviaciones pequeñas de la entrada esperada sólo causarán pequeños inconvenientes en el resultado del análisis, mientras que desviaciones extremas pueden producir grandes inconvenientes.

Un sistema robusto requiere, claramente, métodos que sacrifiquen algo de la tradicional idea de recuperar un análisis completo y exacto. Para evitar la situación donde el analizador puede sólo detener y reportar la falla en el análisis de una entrada, una opción es aliviar algunas de las restricciones gramaticales hasta tal forma que una potencial sentencia no gramatical obtiene un análisis completo.

El problema clave de este enfoque es que, como el número de errores crece, el número de alternativas de alivio de restricciones que

son compatibles con los análisis de toda la entrada pueden salirse de control, y que la búsqueda por la mejor solución sea, por lo tanto, muy difícil de controlar.

Uno puede entonces centrarse en el diseño de la gramática, haciéndola menos rígida con la esperanza de que esto permitirá procesar con más tolerancia. La cantidad de información contenida en las representaciones estructurales producidas por el analizador es usualmente concerniente a la distinción entre análisis a fondo y análisis superficial. Un análisis profundo típicamente captura las dependencias de larga distancia. En contraste, un análisis superficial hace uso de una representación más esquelética. Esto se hace, primero, mediante asignar todas las posibles etiquetas gramaticales y sintácticas a todas las palabras. Aplicando entonces reglas de reconocimiento de patrones para desambiguar las etiquetas y como resultado disminuir el número de análisis.

El resultado constituye una estructura de dependencia en el sentido que esto sólo provee relaciones entre palabras, y puede ser ambiguo debido a que las entidades de dependientes no son completamente especificadas. Una segunda opción es sacrificar las entradas completas, es decir, la cobertura de la entrada, esto mediante pasar solamente los fragmentos que están bien formados de acuerdo con la gramática.

Una tercera opción es sacrificar la noción tradicional de un análisis constructivo, es decir, el análisis de sentencias mediante la construcción de representaciones sintácticas impuestas por reglas de una gramática. En lugar de eso, poder usar un análisis eliminativo, el cual consiste en iniciar primero un máximo de condiciones y después reducir gradualmente los análisis que son ilegales de acuerdo con las restricciones dadas, hasta tener sólo análisis legales.

2.3.4. Desambiguación

Una observación básica es que, aunque una gramática general permitiría un gran número de análisis de casi cualquier sentencia no trivial, muchos de estos análisis serían extremadamente implausibles en el contexto de un dominio particular.

Basados en un corpus suficientemente extenso analizado por la gramática original, es entonces posible identificar combinaciones comunes de reglas de una gramática y guardarlas en un único conjunto de reglas. El resultado sería una gramática especializada, la cual, comparada con la gramática original, tendría un número más grande de reglas, pero una estructura más simple, reduciendo la ambigüedad y permitiendo un rápido procesamiento usando un análisis izquierda-derecha (LR, por sus siglas en inglés: Left-Right).

Otra posibilidad es usar un método híbrido que le dé una jerarquía a un conjunto de análisis con base en su probabilidad en un determinado dominio, esto se basa en datos de un conjunto de entrenamiento. En consecuencia, la desambiguación permite, naturalmente en un sentido u otro, la aplicación de inferencia estadística.

2.4. Análisis semántico

Para textos numerosos, las aplicaciones específicas de PLN para análisis semántico pueden incluir recuperación y extracción de información, generación de resúmenes, minería de datos y traducción automática, entre otras.

El análisis semántico es también muy relevante en el esfuerzo de generar ontologías web y sistemas de representación del conocimiento.

Asimismo, existe un gran valor en llevar a cabo análisis semánticos tan bien como sea posible, los cuales reflejen la realidad cognitiva de los hablantes. Esto hace más fácil modelar la intuición de hablantes nativos y simular sus procesos de inferencia; sin embargo, existe incertidumbre en saber hasta dónde esto es posible.

En la lingüística en general, el análisis semántico se refiere a examinar el significado de las palabras, expresiones fijas y el contexto de los enunciados. En la práctica, esto significa convertir las expresiones originales en algún tipo de metalenguaje semántico.

Muchos enfoques desde la influencia de la lógica psicológica se han orientado al significado condicional de la verdad, pero tales análisis son demasiados limitados para permitir la comprensión completa del uso del lenguaje ordinario o para solventar las diferentes aplicaciones prácticas requeridas, en especial aquellas que involucran una interfaz humano-computadora o de razonamiento ingenuo de usuarios comunes.

Existe una división natural hecha entre semántica léxica, la cual trata con el significado de las palabras y de las combinaciones fijas de éstas, y la semántica supraléxica, la cual tiene que ver con los significados del ampliamente indefinido número de combinaciones de palabras, como frases y sentencias, permitidas por la gramática.

Muchos lingüistas ahora hablan de lexicogramática, en lugar de mantener una distinción estricta entre lexicón y gramática. En parte, esto es debido a que es evidente que la potencial combinatoria de palabras es fuertemente determinada por sus significados, en cierto sentido porque es claro que muchas construcciones gramaticales tienen significados específicos de construcción.

Un punto importante en la teoría semántica es referente a si es posible dibujar una línea bien marcada entre el contenido semántico, en el sentido del contenido codificado en la lexicogramática y el conocimiento enciclopédico general.

En términos generales, la primera evidencia de la semántica lingüística viene de las interpretaciones de los hablantes nativos del uso de las expresiones lingüísticas en contexto, de la observación natural del lenguaje en uso, y de la distribución de expresiones lingüísticas; esto último se refiere al patrón de uso, colocación y frecuencia.

Un requerimiento frecuentemente identificado para el análisis semántico en el PLN se trata de la resolución de ambigüedad. Desde el punto de vista del procesamiento computacional de textos, muchos enunciados hechos por humanos están abiertos a múltiples interpretaciones, debido a que las palabras pueden tener más de un significado (ambigüedad léxica), o debido a que ciertas palabras, tal como los cuantificadores, adverbios de modo u operadores negativos pueden darle un significado no determinista a una oración (ambigüedad de alcance), o debido a que la referencia pretendida de los pronombres puede no ser clara (ambigüedad de referencia).

En relación con las ambigüedades léxicas, es usual distinguir entre la homonimia, la cual consiste en la relación de semejanza en la manera de escribirse o pronunciarse, es decir, la que presentan dos palabras de significado diferente; y la polisemia, cuando una palabra tiene más de un significado. Otros problemas para la semántica léxica se plantean por la existencia de expresiones figurativas o las unidades multipalabra; estas últimas son palabras cuyo significado no es predecible mediante el significado de las palabras individuales.

2.4.1. Teorías y enfoques de la representación semántica

Diferentes teorías y enfoques de la representación semántica pueden ser posicionados en dos categorías: formal *vs.* cognitiva y composicional *vs.* léxica.

Las teorías formales han sido fuertemente definidas desde finales de 1960, mientras que los enfoques cognitivos se han vuelto populares en las últimas tres décadas, movidas bajo la influencia de la ciencia cognitiva y la psicología.

Por una parte, la semántica composicional tiene que ver con la construcción, de lo menor a lo mayor, del significado, empezando con los componentes léxicos. Por otra parte, la semántica léxica tiene como objetivo analizar el significado de los componentes léxicos por medio de analizar su estructura interna y contenido o por medio de representar las relaciones con otros elementos en el lexicón.

2.4.2. Enfoques lógicos

Los enfoques lógicos del significado generalmente se centran en problemas composicionales, desde la hipótesis que el significado de expresiones supraléxicas está determinado por el significado de sus componentes y la forma en que estas se combinan.

Diferentes sistemas lógicos han sido y están siendo desarrollados en el área de la semántica lingüística y el PLN. Uno de esos sistemas es *predicate logic*, en el cual las propiedades de conjunto de objetos pueden ser expresadas por medio de predicados, conectivas lógicas y cuantificadores. Esto se hace promoviendo la sintaxis y especificando cómo los elementos del lenguaje lógico pueden ser combinados para

formar expresiones lógicas correctas, y mediante proveer semántica especificando qué significan tales expresiones dentro del sistema lógico.

Aunque algunas veces se presentó como una teoría general de conocimiento, *predicate logic* no es suficientemente robusto como para representar las complejidades del significado semántico y es fundamentalmente diferente del razonamiento humano.

2.4.3. Teoría de representación de discurso

La teoría de representación de discurso fue desarrollada a principios de los 80 por Kamp (1981) para la captura de la semántica de los discursos o textos, es decir, las secuencias coherentes de las sentencias o enunciados. La idea básica es que a medida que se desarrolla un discurso o texto el oyente construye una representación mental, representada por la llamada estructura de representación del discurso, y que por cada sentencia entrante solicita adiciones a esta representación. De esta manera, resulta un enfoque dinámico para la semántica del lenguaje.

La teoría de representación de discurso requiere los siguientes componentes: una definición formal de la representación del lenguaje que consiste en una definición recursiva del conjunto de todas las estructuras de representación del discurso bien formadas y una semántica modelo-teórica para los miembros de este conjunto; un procedimiento de construcción que especifique cómo una estructura de representación del discurso puede ser agregada cuando se tiene nueva información.

El enfoque de la teoría de representación de discurso está bien adaptado para tratar, por ejemplo, con resolución de anáfora.

2.4.3.1. Lexicón generativo de Pustejovsky

Otro enfoque dinámico de la semántica, pero que se enfoca en elementos léxicos, es la teoría del lexicón generativo de Pustejovsky (1991).

Dicho autor postula que, dentro de contextos particulares, los componentes léxicos asumen diferentes sentidos. Él desarrolló la idea de un lexicón en el cual los sentidos de las palabras en el contexto pueden ser flexiblemente derivados con base en una representación multinivel que involucra al menos los siguientes niveles:

1. Estructura del argumento: especificación del número y tipo de los argumentos lógicos y cómo ellos se desenvuelven sintácticamente.

2. Estructura de evento: definición del tipo de evento de un elemento léxico y una frase. El tipo de evento incluye estados, procesos y transiciones; la estructuración de subeventos es posible.

3. Estructura de cualidades subjetivas: son modos de explicación, que comprenden las cualidades subjetivas en cuatro tipos: constitutiva, ¿de qué está hecho el objeto?; formal, ¿qué es el objeto?, ¿cómo se distingue en un amplio dominio?; de fin definido, ¿cuál es el propósito o función del objeto?; y de naturaleza, ¿cómo surgió el objeto?, factores que influyeron en su surgimiento.

4. Estructura de herencia léxica: identificación de cómo una estructura léxica está relacionada con otras estructuras en el lexicón y su contribución con la organización global del lexicón.

Los procesos generativos del sistema de Pustejovsky interconectan los niveles de representación con el objetivo de proveer una interpretación

composicional de los elementos léxicos en el contexto. El lexicón generativo tiene diferentes propósitos en los enfoques lógico-inspirados para la semántica. Se orienta a un enfoque léxico descomposicional detallado para la semántica lingüística, y al mismo tiempo provee herramientas que permiten conocer el significado en el contexto mediante un cálculo composicional.

2.4.3.2. Metalenguaje semántico natural

El metalenguaje semántico natural es un sistema descomposicional basado en primas semánticas establecidas empíricamente, es decir, significados simples e indefinibles que parecen estar presentes como significados de palabra en todos los lenguajes.

El sistema de metalenguaje semántico natural usa un metalenguaje, el cual es esencialmente un subconjunto estandarizado del lenguaje natural. Éste consiste en un subconjunto que contiene palabras con su significado, junto con un subconjunto con sus propiedades sintácticas asociadas.

El modo formal de la representación del significado en el enfoque de metalenguaje semántico natural es la explicación semántica. Esto es una paráfrasis reductiva, es decir, un intento de expresar en otras palabras lo que el hablante está diciendo cuando pronuncia la expresión que es explicada. Originado con Wierzbicka (1972), el metalenguaje semántico natural ha sido desarrollado y refinado por 35 años; sin embargo, aunque este enfoque es posiblemente la mejor teoría de semántica léxica en la escena contemporánea, tiene pocas aplicaciones para el PLN.

Los investigadores de este enfoque reconocen que, para muchas palabras en un vocabulario concreto, no es posible producir explicaciones plausibles directamente en términos de primas semánticas únicas. En

vez de eso, las explicaciones típicamente requieren una combinación de primas semánticas y de ciertos significados léxicos conocidos en la teoría del metalenguaje semántico natural como moléculas semánticas. Las moléculas semánticas funcionan como bloques de construcción para lograr conceptos más complejos. Estas moléculas también se pueden anidar, una dentro de otra, creando cadenas de dependencia semántica.

2.4.3.3. Semántica orientada a objetos

La semántica orientada a objetos es relativamente un nuevo campo en la semántica lingüística. Aunque es bastante restringida en los dominios de aplicaciones semánticas, ésta es principalmente aplicada a la representación de significado verbal.

La motivación básica detrás del esparcimiento del paradigma computacional orientado a objetos en la semántica lingüística es su accesibilidad intuitiva. El sistema cognitivo humano se basa en entidades y en qué son ellas, cómo se relacionan con otras entidades, qué les pasa y qué hacen y cómo ellas interactúan con otras. Esto corresponde al enfoque orientado a objetos, en el cual el concepto de objeto es central, cuyas características, relaciones con otras entidades, comportamiento e interacciones son modeladas de forma rigurosa.

2.4.3.4. Hechos relacionales y ontologías

Las relaciones semánticas entre elementos léxicos (relaciones de sentido) generan la base para las redes de palabras, tal como la base de datos digital llamada WordNet (Fellbaum, 1998) y de enfoques similares para otros lenguajes diferentes al inglés, por ejemplo, la base de datos multilenguaje EuroWordNet (Vossen, 1998, 2001). El uso de ontologías

y la estructura lingüística se está convirtiendo en un área sobresaliente en la representación de conocimiento y el PLN.

Las relaciones de sentido pueden ser vistas como un revelador de la estructura semántica de un lexicón. Existen relaciones de sentido horizontales y verticales. Las relaciones horizontales incluyen la sinonimia, es decir, palabras con el mismo significado, pero con diferente forma lingüística. También existen dos relaciones verticales principales que incluyen la hiponimia y la meronimia. La primera ocurre cuando el significado de un elemento léxico, el hipónimo, es más específico que el significado del otro, el hiperónimo; la segunda ocurre cuando el significado de un elemento léxico especifica que forma parte de otro elemento léxico, por ejemplo, dedo es merónimo de mano.

Las ontologías han sido desarrolladas y empleadas en el área de inteligencia artificial y representación de conocimiento y más generalmente en las ciencias de la computación.

Capítulo 3

Métodos para la identificación del engaño

3.1. Antecedentes

La detección de texto engañoso mediante algoritmos de aprendizaje automático trata de identificar las características que definen a estos textos de acuerdo con un conjunto previo de datos llamado conjunto de entrenamiento. De esta manera, los algoritmos pueden analizar un nuevo texto (no contenido en el conjunto de entrenamiento) e identificar si pertenece a alguna clase específica.

Los investigadores han estudiado la detección de engaño mediante el uso de diferentes fuentes de características. Algunos de estos métodos están basados en herramientas de bolsa de palabras (BoW), en las cuales no se considera el orden de las palabras (contexto); otros añaden información sintáctica como características. En algunos casos, se buscan algunas señales generales de engaño (DePaulo *et al.*, 2003), tales como el uso de palabras únicas, autoreferencias o modificadores, entre otros.

En general, los enfoques de estilo lingüístico (como ns-gramas) analizan las relaciones entre las palabras; en cambio, los enfoques de BoW (LDA, MEP, LIWC, n-gramas) ignoran la gramática e incluso el orden

de las palabras y se basan principalmente en el conteo de instancias de cada palabra.

Para detectar textos engañosos, una técnica comúnmente aplicada es usar n-gramas. Este método puede extraer características de un texto basado en diferentes elementos; por ejemplo, palabras, sílabas, fonemas, letras, entre otros.

En el trabajo de Fusilier *et al.* (2015) los n-gramas de palabra se compararon con los n-gramas de carácter. Estos últimos han demostrado tener un mejor rendimiento en el conjunto de datos OpSpam (descrito en la sección 4.2.2). Aunque los n-gramas logran resultados aceptables por sí mismos, generalmente se complementan con otras técnicas del PLN debido a que la combinación de características ha demostrado mejorar los resultados.

Otro enfoque de BoW consiste en utilizar el diccionario Linguistic Inquiry and Word Count (LIWC), que incluye una clasificación de palabras y una herramienta de recuento. Newman *et al.* (2003), por ejemplo, al analizar las categorías de palabras de LIWC, encontraron que los mentirosos usan menos autoreferencias y usan palabras de emociones negativas. Este trabajo sentó las bases para que la herramienta LIWC sea ampliamente utilizada por otros investigadores (Schelleman Offermans y Merckelbach, 2010; Toma y Hancock, 2012).

Hauch *et al.* (2012) introdujeron un metaanálisis de varios trabajos de investigación de identificación de textos engañosos. Éste se centró en categorías lingüísticas específicas, por ejemplo, las contenidas en LIWC. Los hallazgos de la investigación sugieren que los mentirosos usan ciertas categorías lingüísticas en una tasa diferente que los que dicen la verdad.

La detección de engaño se ha aplicado en diversas situaciones particulares. En la investigación de Williams *et al.* (2014), se hizo una comparación entre las mentiras dichas por niños y las mentiras dichas por adultos. Ésta se llevó a cabo con el objetivo de detectar el engaño en los tribunales donde los niños testifican. Los autores eligieron 48 niños y 28 adultos para generar un conjunto de datos; la mitad de los niños y adultos contaron mentiras y la mitad de ellos dijeron la verdad. De esta forma, se utilizó la herramienta LIWC para generar las muestras para la clasificación. Los resultados mostraron que existen diferencias significativas entre los textos verdaderos y los falsos, principalmente en variables lingüísticas tales como las autoreferencias singulares (por ejemplo, *I*, *my*, *me*), autoreferencias (por ejemplo, *we*, *our*, *us*) y las emociones negativas. Además, los resultados de la investigación mostraron que las variables lingüísticas se encontraron en distintas proporciones dependiendo de si la mentira fue contada por un niño o por un adulto.

Los estudios desarrollados mediante el uso de herramientas de BoW han tenido éxito; sin embargo, en un esfuerzo por mejorar los resultados, se ha tenido en cuenta el contexto de las oraciones, al analizar las relaciones sintácticas de las palabras con el uso de árboles de dependencia (Xu y Zhao, 2012). En general, el uso de las relaciones sintácticas no ha mostrado un rendimiento sobresaliente en la tarea de clasificar texto engañoso. Aunque complementar este método con un enfoque de BoW puede mejorar los resultados.

En el estudio de Pérez Rosas y Mihalcea (2014b), las características se generaron utilizando diferentes enfoques (por ejemplo, etiquetas gramaticales (POS), gramáticas libres del contexto (CFG), unigramas,

LIWC y combinaciones de éstos). Los autores predijeron, con una exactitud de entre 60 % y 70 %, si una persona de sexo femenino o masculino había escrito un texto engañoso. En los resultados mostrados, el uso de POS y CFG no produjo una mejora significativa en la exactitud con respecto a los unigramas y LIWC. Esto sugiere que los enfoques de BoW tienen un desempeño similar a los enfoques de estilo lingüístico.

Para probar la eficiencia de los algoritmos de aprendizaje se recurre a conjuntos de datos llamados corpus (singular) o corpora (plural). Los conjuntos de datos se pueden dividir en dos subconjuntos: el ya mencionado conjunto de entrenamiento que le da el conocimiento previo al clasificador; y el conjunto de prueba, el cual consiste en documentos que se clasificarán con base en el conocimiento previo.

3.2. Detección de engaño en tópicos controversiales

En el trabajo de investigación de Mihalcea y Strapparava (2009), se analizó la detección de texto engañoso y texto veraz mediante el enfoque de aprendizaje automático. De esta manera, se pretendían capturar características subyacentes, de forma automática en el texto que permitieran identificar si un texto es engañoso o veraz. Los autores se basaron solamente en la obtención de características lingüísticas, aunque otros trabajos incluyen características no lingüísticas (por ejemplo, características acústicas), esto debido a que la mayor parte de información que se encuentra en la web son textos.

El objetivo de la investigación fue saber si los textos engañosos y veraces realmente eran separables, es decir, si existen características que sólo se pueden encontrar o que se encuentran con mayor probabilidad

en los textos engañosos. Para resolver lo anterior, los autores crearon un conjunto de datos de tópicos controversiales; este conjunto contiene textos cortos engañosos y veraces.

De forma específica, el conjunto de datos que generaron (detallado en las sección 4.2.3) consta de tres tópicos: *aborto*, *pena de muerte* y mejores amigos. Como en todo caso, y para poder evaluar el enfoque propuesto, fue necesario que el conjunto de datos fuera etiquetado, es decir, que cada texto tenga su respectivo identificador para saber si es engañoso o veraz. Para realizar la labor de etiquetado optaron por usar el servicio del turco mecánico de Amazon, el cual es una herramienta web que permite a diferentes usuarios del sitio ayudar en una tarea a cambio de una aportación monetaria.

El conjunto se formó mediante dos dinámicas. En primer lugar, en el caso de los tópicos acerca de *aborto* y *pena de muerte* la tarea consistió en que los usuarios debieron escribir su opinión real acerca de cada uno de estos dos tópicos; en seguida se les pidió escribir la opinión contraria a su opinión real. De esta forma, el resultado es un texto con información veraz y un texto con información engañosa. En segundo lugar, para el caso del tópico referente a los mejores amigos, se les pidió a los usuarios que pensaran en su *mejor amigo* y describieran las razones de su amistad; después se les pidió que pensaran en una persona a la que no pudieran soportar y la describieran como si fuera su *mejor amigo*.

Como resultado, en el primer caso, los participantes mienten acerca de sus creencias, mientras que en el segundo caso los participantes mienten acerca de sus sentimientos hacia otra persona. Finalmente, fueron recolectados 100 textos veraces y 100 engañosos por tópico, sumando un total de 600 textos.

Con el propósito de obtener las características más relevantes del texto engañoso y del texto veraz idearon un método para valorar en qué medida las palabras se inclinan hacia una u otra clase. Tal método se basa en las siguientes medidas: la cobertura del texto engañoso (fórmula 3.1), la cual obtiene la fracción del número de *tokens* de la palabra W_i en el corpus de textos engañosos (D) entre el total de *tokens* del corpus D. De la misma forma, la cobertura del texto veraz (fórmula 3.2) realiza el mismo cálculo, pero esta vez para el corpus de texto veraz (T).

Finalmente, se calcula el valor de dominancia (fórmula 3.3) de la clase C (clases incluidas en LIWC, en su versión 2001), el cual es la proporción entre la cobertura del texto veraz y la cobertura del texto engañoso, calculado para la misma clase.

Los tres tópicos fueron mezclados para formar dos conjuntos de datos generales: el conjunto T y el conjunto D, de texto veraz y engañoso, respectivamente. Sobre estos conjuntos generales fueron calculados los valores de dominancia de cada clase C. De esta manera, por una parte, un resultado cercano a 1 indicará que existe una distribución similar de palabras en la clase C tanto de texto veraz como de texto engañoso. Por otra parte, si el resultado es menor a 1, significará que la clase es predominante del texto veraz, en cambio, si el resultado es superior a 1, significará que la clase es predominante del texto engañoso. En la tabla III.1 se muestran algunos valores de clases dominantes obtenidos por los autores.

En los experimentos se clasificaron los textos mediante un clasificador naïve Bayes (NB) y una máquina de vectores de soporte (SVM, por sus siglas en inglés: Support Vector Machine). Primero, se clasificaron los tres tópicos por separado; los resultados son mostrados en la tabla III.2,

en la cual se especifican los valores de exactitud obtenidos por cada clasificador; asimismo, la exactitud promedio de los tres tópicos. Se observa también que el mejor resultado en promedio fue obtenido por el clasificador NB.

Siguiendo el trabajo de Mihalcea y Strapparava (2009), también se realizó un estudio para detectar texto engañoso escrito en español: Almela *et al.* (2012) recopiló un nuevo conjunto de datos con temas que abarcan la adopción homosexual, opiniones sobre la tauromaquia y sentimientos acerca de un *mejor amigo*. Cien documentos falsos y cien verdaderos fueron recopilados para cada tema con un promedio de 80 palabras por documento. Distintas dimensiones de LIWC se utilizaron para lograr una clasificación más precisa por medio de una máquina de vectores de soporte (SVM).

$$Cobertura_D(C) = \frac{\sum_{W_i \in C} Frecuencia(W_i)}{Tamaño_D} \tag{3.1}$$

$$Cobertura_T(C) = \frac{\sum_{W_i \in C} Frecuencia(W_i)}{Tamaño_T} \tag{3.2}$$

$$Dominancia_D(C) = \frac{Cobertura_D(C)}{Cobertura_T(C)} \tag{3.3}$$

3.3. Detección de engaño en opiniones sobre hoteles

El objetivo de la investigación de Ott *et al.* (2011) fue identificar las opiniones engañosas que se definen como opiniones ficticias que han sido deliberadamente escritas para sonar auténticas.

Tabla III.1: Clases de LIWC dominantes del texto engañoso y veraz

Clase	Puntaje	Ejemplo de palabras
Texto engañosos		
Metáfora	1.71	*god, die, sacred, mercy, sin, dead,hell, soul, lord*
Tú/Ustedes	1.53	*you, thou*
Otros	1.47	*she, her, they, his, them, him, herself, himself, themselves*
Humanos	1.31	*person, child, human, baby, man, girl,humans, individual, male, person, adult*
Certero	1.24	*always, all, very, truly, completely, totally*
Texto veraz		
Óptimo	1.71	*best, ready, hope, accepts, accept, determined, accepted, won, super*
Yo	1.53	*I, myself, mine*
Amigos	1.47	*friend, companion, body*
Sí mismo	1.31	*Our, myself, mine, ours*
Visión	1.24	*Believe, think, know, see, understand, found, thought, feels, admit*

Tabla III.2: Exactitud obtenida por tópico usando una validación cruzada de 10 pliegues

Tópico	Naïve Bayes (%)	SVM (%)
Aborto	70.0	67.5
Mejor amigo	67.4	65.9
Pena de muerte	75.0	77.0
Promedio	70.8	70.1

Para alcanzar este objetivo, los autores exploraron tres enfoques, el primero fue ver el problema como una tarea de categorización de texto mediante el uso de n-gramas y clasificadores, esto con el fin de etiquetar los documentos como veraces o engañosos. El segundo fue optar por un enfoque psicolingüista, en el cual se pretendía verificar cuales son los estados psicológicos, como las emociones negativas y positivas, relacionados con el engaño. Por último, en el tercero se abordó la tarea como un problema de género mediante el cual se percibe la escritura de textos engañosos como de tipo imaginativa y la escritura de textos veraces como de tipo informativa.

Mediante la clasificación de un conjunto de 400 textos engañosos y 400 textos veraces los autores identificaron que los clasificadores de aprendizaje automático, entrenados con características obtenidas mediante enfoques psicolingüistas y de identificación de género, sobrepasan estadísticamente el nivel de los enfoques de categorización de texto, tal como los n-gramas.

Además, remarcaron que las capacidades humanas para diferenciar entre un texto engañoso y uno veraz, como se muestra en la tabla III.3, son limitadas. Esto lo ratificaron mediante una clasificación, hecha por tres participantes, de un subconjunto de 160 documentos. En la tabla también se muestran medidas menos simplistas que la exactitud, éstas son: precisión (P), exhaustividad (R) y medida F (F).

Los autores también dan importancia a que la identificación de texto tiene que ser afrontada mediante el contexto y la motivación que condujo al engaño y no sólo mediante la identificación de un conjunto de señales universales de engaño.

Por una parte, proponen generar características basadas en etiquetas gramaticales (POS tags, por sus siglas en inglés: Part-Of-Speech tags); esto, con base en que algunas investigaciones (Douglas y Broussard, 2000; Rayson *et al.*, 2002) han probado que la frecuencia y distribución de las etiquetas gramaticales tienen cierta relación con la identificación de género. De esta manera, los autores construyen vectores de características basadas en la frecuencia de cada etiqueta gramatical.

Por otra parte, y para comparar los resultados del enfoque que propusieron, usaron LIWC en su versión 2007, generando una característica por cada una de las 80 dimensiones incluidas en este diccionario. Finalmente, con respecto a las fuentes de características, usaron n-gramas debido a que mediante ellos se puede capturar el contenido y contexto de los documentos. Asimismo, generaron características a partir de unigramas, bigramas y trigramas.

Para realizar la clasificación de documentos usaron un clasificador naïve Bayes y una SVM con una validación cruzada de cinco pliegues. Los resultados se muestran en la tabla III.4, en la cual podemos observar que la combinación de características de LIWC y bigramas, clasificados con una SVM, obtuvieron el mejor resultado. Además, con respecto a exactitud y medida F, todas las características generadas y clasificadas con aprendizaje automático superan la clasificación humana (tabla III.3).

Aunque los autores proponen la identificación de texto mediante un enfoque de identificación de género, se puede observar en la tabla III.4 que la generación de características mediante la frecuencia de las etiquetas gramaticales no fue la mejor opción para producir vectores suficientemente informativos. Puesto que los autores argumentan que en los experimentos se probaron todas las combinaciones de fuentes

de características, y dado que no se muestran más combinaciones del enfoque POS, se puede inferir que combinar el enfoque de etiquetas gramaticales con otro enfoque resultó perjudicial para la clasificación.

Tabla III.3: Exactitud, precisión (P), exahustividad (R), y medida F (F) de la identificación de texto engañoso realizada por humanos

		Veraz			Engañoso		
Participante	Exactitud	P	R	F	P	R	F
Juez 1	61.9	57.9	87.5	69.7	74.4	36.6	48.7
Juez 2	56.9	53.9	95.0	68.8	78.9	18.8	30.3
Juez 3	53.1	52.3	70.0	59.9	54.7	36.3	43.6

Tabla III.4: Resultados de la aplicación de diferentes fuentes de generación de características y aplicación de dos clasificadores

		Veraz			Engañoso		
Características	Exactitud	P	R	F	P	R	F
Identificación de género							
POS/SVM	73.0	75.3	68.5	71.7	71.1	77.5	74.2
Enfoque psicolingüístico							
LIWC/SVM	75.8	77.2	76.0	76.6	76.4	77.5	76.9
Categorización de texto							
Unigramas/SVM	88.4	89.9	86.5	88.2	87.0	90.3	88.6
Bigramas/SVM	89.6	90.1	89.0	89.6	89.1	90.3	89.7
LIWC+bigramas/SVM	89.8	89.8	89.8	89.8	89.8	89.8	89.8
Trigramas/SVM	89.0	89.0	89.0	89.0	89.0	89.0	89.0
Unigramas/NB	88.4	92.5	83.5	87.8	85.0	93.3	88.9
Bigramas/NB	88.9	89.8	87.8	88.7	88.0	90.0	89.0
Trigramas/NB	87.6	87.7	87.5	87.6	87.5	87.8	87.6

3.4. Detección de engaño en declaraciones

En la investigación de Fornaciari y Poesio (2012) se analizó la identificación de texto engañoso mediante la creación de subconjuntos a partir de un conjunto general o corpus. La agrupación de subconjuntos homogéneos de sujetos se da, por ejemplo, con la separación de género, es decir, si el sujeto que escribió el texto es hombre o mujer. La agrupación de conjuntos fue realizada tanto de forma automática, mediante técnicas de agrupamiento automático y también mediante el uso de los metadatos contenidos en los conjuntos de datos. De esta manera, las agrupaciones, como argumentan los autores, podrían generar la eficiencia en la detección de texto engañoso, basados en que el comportamiento similar en la escritura de los autores podría influir en la detección del engaño.

En este estudio se generó un conjunto llamado DeCour (Deception in Court), el cual es un conjunto de datos de transcripciones de 35 audiencias llevadas a cabo en cuatro cortes de Italia. Los textos se componen de preguntas hechas en la corte a los testigos que defienden al acusado, como resultado el conjunto de datos contiene etiquetas que informan si la oración o respuesta transcrita es veraz o engañosa.

Los autores crearon vectores de características con tres tipos de características: la primera agrega información básica como el tamaño de las sentencias y el número de palabras que tienen más de seis caracteres; la segunda agrega información léxica mediante el uso de LIWC 2001, agregando al vector de características las 80 dimensiones con las que cuenta dicha versión; la última agrega frecuencias de n-gramas de lemas y de etiquetas gramaticales. Los autores tomaron en cuenta, en los experimentos, desde unigramas hasta pentagramas.

Mediante la obtención de n-gramas, con valores de n desde 1 hasta 5, se generaron dos listas de los n-gramas sobresalientes, una correspondiente al texto engañoso y otra al texto veraz. Dichas listas se compararon para evitar que una característica igual se presente en ambas; de esta manera, se puede generar solamente información particular que permita una clasificación eficiente. Tres experimentos fueron realizados usando el conjunto de datos creado. El primero consistió en usar todo el conjunto de datos por medio de establecer un conjunto de entrenamiento y de prueba.

En el segundo se generaron subconjuntos del corpus completo, de forma que se eliminaron aquellos vectores que estuvieran demasiado alejados de los grupos (*outliers*). Estos grupos fueron establecidos mediante un método no supervisado de agrupamiento de patrones.

En el tercero se hizo uso de los metadatos que se recabaron con el corpus. En principio éste cuentó con la siguiente información: el género (masculino o femenino), el lugar de nacimiento y la edad al momento de la declaración. Finalmente, los autores decidieron tomar en cuenta sólo el género de los sujetos, en específico, sólo aquellos participantes de género masculino. Esta restricción impactó fuertemente sobre el conjunto de entrenamiento y no a tal grado el conjunto de prueba.

El conjunto de prueba consta de 426 oraciones de las cuales 190 son engañosas y 236 veraces. Los resultados de la clasificación en los tres diferentes experimentos se muestra en la tabla III.5. Se puede observar, por una parte, que el peor resultado, respecto al promedio de la medida F, se obtiene cuando sólo se contempla el género masculino, esto podría deberse a que el conjunto de entrenamiento se redujo drásticamente.

Por otra parte, el mejor resultado se obtuvo en el experimento donde se eliminaron los *outliers*, lo que pudo permitir un grupo más homogéneo.

Tabla III.5: Resultados de la clasificación de corpus DeCour. Se muestran: precisión (P), exhaustividad (R) y medida F (F), asimismo el promedio de medidas F

		Veraz			Engañoso		
Experimento	Promedio F	P	R	F	P	R	F
Todo el corpus	60.1	62.9	94.0	75.3	80.8	31.0	44.9
Sin *outliers*	64.0	66.8	93.8	77.9	81.0	36.2	50.0
Género masculino	59.5	67.8	94.2	78.9	74.4	27.4	40.0

3.5. Detección de engaño en mezcla de dominios

En la sección 3.4 se consideró que la creación de subconjuntos basados en el género y en los *outliers* tendría cierto impacto en la eficiencia de la clasificación. Asimismo, en la investigación de Pérez Rosas y Mihalcea (2014a) se realizan experimentos con textos escritos por personas de diferentes países para verificar hasta qué punto es posible identificar el engaño, por ejemplo, usando los textos de un país para identificar engaño en los textos de otro país diferente.

La mayoría de estudios de detección de engaño están enfocados a documentos de determinado país e idioma, principalmente el idioma inglés. En este sentido, los autores argumentan que los textos, principalmente aquellos que se encuentran en los sitios web, tienden, en cierta proporción, a ser escritos por personas que tienen diferente cultura o idioma; esto implica que las personas pueden tener diferentes creencias y valores morales. En consecuencia, se pone en tela de juicio si los enfoques que se basan en un sólo idioma y cultura podrán ser

igualmente aplicados a una mezcla de textos interculturales e incluso de diferente idioma (este último, usando herramientas de traducción).

En este estudio se analizó la detección de engaño con textos escritos en tres diferentes culturas: Estados Unidos, México y la India. Por lo cual, los autores generaron tres conjuntos de datos que abordan, para cada uno, tres tópicos controversiales: *aborto*, *pena de muerte* y mejores amigos. Lo anterior tal y como se manejó en el estudio presentado en la sección 3.2, excepto para los textos recabados en español de México; estos últimos se obtuvieron mediante una página creada por los autores debido a la poca participación recibida mediante el turco mecánico del sitio de internet de Amazon.

Las fuentes de generación de características para este estudio fueron dos. Como en la mayoría de las investigaciones mostradas en esta sección (estado del arte), los unigramas son requeridos para generar una parte del vector de características, por lo que se puede inferir que las palabras por sí solas son una importante fuente informativa para detectar el engaño.

Para complementar el vector de características, se usó una segunda herramienta que se ha vuelto muy popular en la detección de texto engañoso, ésta es LIWC, que como se ha dicho antes es un enfoque que brinda un análisis psicolingüístico.

Se realizaron dos tipos de experimentos: el primero es una clasificación sobre los conjuntos de datos con documentos del mismo país; el segundo se trata de un cruce de dominios, es decir, tomar dos conjuntos (de los tres disponibles) para formar el conjunto de entrenamiento y usar el conjunto restante para generar el conjunto de prueba.

Este último experimento se hizo tanto de tópico mezclado, donde se aplica el dominio cruzado sobre los tres tópicos sobre cada cultura

Tabla III.6: Experimentos realizados sobre los corpora de cada país, por separado

Tópico	LIWC	Unigramas	
		En dominio	Dominio cruzado
Inglés: EUA			
Aborto	73.0	63.8	80.4
Mejor amigo	73.0	74.5	60.8
Pena de muerte	58.1	58.1	77.2
Inglés: India			
Aborto	56.0	46.0	50.0
Mejor amigo	71.4	60.5	57.2
Pena de muerte	63.5	57.5	54.0
Español: México			
Aborto	62.2	52.5	57.7
Mejor amigo	75.3	66.7	50.5
Pena de muerte	62.2	54.9	63.4

por separado; como para cultura cruzada, donde se aplica el dominio cruzado sobre las tres culturas. Este último experimento se llevó a cabo con el objetivo de probar si las características que sirven para identificar texto engañoso en un determinado lenguaje de un país, son equivalentes o útiles al buscar engaño en el lenguaje manejado por otro país diferente (de diferente cultura).

En la tabla III.6 se muestran los resultados de la clasificación, de forma individual, de cada cultura mediante la generación de características con el enfoque psicolingüístico y mediante el uso de unigramas. Sobre este último se aplica la clasificación en dominio, en la cual se clasifica cada tópico por separado (*aborto*, *mejor amigo* y *pena de muerte*) y la clasificación de dominio cruzado, en la cual se usan dos tópicos como entrenamiento y un tercer tópico como prueba.

Tabla III.7: Clasificación de cultura cruzada con LIWC y unigramas como fuente de características

Tópico	LIWC	Unigramas
Entrenamiento: Inglés: EUA, prueba: Inglés: India		
Aborto	52.3	57.9
Mejor amigo	59.5	51.0
Pena de muerte	53.5	59.0
Entrenamiento: Inglés: India, prueba: Inglés: EUA		
Aborto	62.5	55.5
Mejor amigo	55.8	53.2
Pena de muerte	39.2	50.7
Entrenamiento: Inglés: EUA, prueba: español: México		
Aborto	53.9	61.5
Mejor amigo	67.7	65.0
Pena de muerte	62.2	59.8
Entrenamiento: Inglés: India, prueba: español: México		
Aborto	43.6	55.1
Mejor amigo	60.8	67.2
Pena de muerte	59.8	51.2

En la tabla III.7 se muestran los resultados de la clasificación de dominio cruzado mediante el uso de dos lenguajes como conjunto de entrenamiento y el lenguaje restante como conjunto de prueba.

En la tabla III.8 se muestran las clases predominantes tanto del texto engañoso como del texto veraz. Además, algunos ejemplos de palabras incluidas en cada clase. Similar al trabajo mostrado en la sección 3.2, un puntaje significativamente menor a 1 indica que la clase es predominante del texto veraz, mientras que un valor significativamente por encima de uno indica que la clase es predominante del texto engañoso.

3.6. Detección de engaño en opiniones sobre libros

Las opiniones engañosas en la venta de libros, de acuerdo con el estudio de Fornaciari y Poesio (2014), ocurren cuando los autores escriben una opinión brillante sobre sus propios libros.

Con el propósito de analizar las opiniones engañosas presentes en la venta de libros a través de Amazon, los autores crearon un nuevo conjunto de datos que etiquetaron con base en algunas señales de engaño, la presencia de dichas señales determinó si la opinión fue considerada engañosa o veraz. El enfoque usado para el etiquetado fue el aprendizaje de multitudes propuesto por Raykar *et al.* (2010); además se evaluó la efectividad de diferentes métodos de etiquetado de acuerdo con el rendimiento de los modelos generados para detectar opiniones engañosas.

Los autores argumentan que la gran desventaja de otros trabajos que tratan la detección de engaño es que son carentes de un conjunto de datos con textos reales; en cambio, son recreados artificialmente; por ejemplo, cuando se les pide a los participantes mentir acerca de una opinión real.

El punto es que, al darle permiso a una persona de mentir, es posible que las características de engaño no se reflejen en el texto, o se reflejen de forma parcial; incluso cabe la posibilidad de que lo que se esté capturando no sea el mismo fenómeno; por lo tanto, la identificación y uso del modelo no será congruente con los textos engañosos en fenómenos reales.

Por lo anterior, los autores propusieron un nuevo método para identificar opiniones engañosas en Amazon. Este sistema se basa en dos procesos que se definen a continuación:

En el primero se colaboró con un experto en detección de opiniones engañosas para identificar una serie de criterios para encontrar opiniones

Tabla III.8: Clases de LIWC predominantes en el texto engañoso y en el texto veraz

Clase	Puntaje	Ejemplo	Clase	Puntaje	Ejemplo
Inglés: EUA					
Engañoso			Veraz		
Metáfora	1.77	*Die, died, hell, sin, lord*	Visión	0.68	*Accept, believe, understand*
Otros	1.46	*He, her, herself, him*	Yo	0.66	*I, me, my, myself*
Tú/Ustedes	1.41	*Thou, you*	Optimismo	0.65	*Accept, hope, top, best*
Otras referencias	1.18	*He, her, herself, him*	Nosotros	0.55	*Our, ourselves, us, we*
Emociones negativas	1.18	*Afraid, agony, awful, bad*	Amigos	0.46	*Buddies, friend*
Inglés: India					
Engañoso			Veraz		
Negación	1.49	*Cannot, neither, no, none*	Pasado	0.78	*Happened, helped, liked, listened*
Físico	1.46	*Heart, ill, love, loved*	Yo	0.66	*I, me, mine, my*
Futuro	1.42	*Be, may, might, will*	Optimismo	0.65	*Accept, accepts, best, bold*
Otros	1.17	*He, she, himself, herself*	Nosotros	0.55	*Our, ourselves, us, we*
Humanos	1.08	*Adult, baby, children, human*	Amigos	0.46	*Buddies, companion, friend, pal*
Español: México					
Engañoso			Veraz		
Certeza	1.47	Jamás(*never*), siempre(*always*)	Optimismo	0.66	Aceptar(*accept*), animar(*cheer*)
Humanos	1.28	Bebé(*baby*), persona(*person*)	Sí mismo	0.65	Conmigo(*me*), tengo(*have*), soy(*am*)
Tú/Ustedes	1.26	Eres(*are*), estás(*be*), su(*his/her*)	Nosotros	0.58	Estamos(*are*), somos(*be*), tenemos(*have*)
Negación	1.25	Jamás(*never*), tampoco(*neither*)	Amigos	0.37	Amigo(*friend*), amistad(*friendship*)
Otros	1.22	Es(*is*), está(*are*), otro(*other*)	Pasado	0.32	Compartimos(*share*), vivimos(*lived*)

no genuinas. De algunas de estas opiniones ya se tenía, de antemano, la seguridad de que eran engañosas debido a que se obtuvieron confesiones de personas dedicadas a la escritura de éstas.

En el segundo, los autores desarrollaron un enfoque que identificó la veracidad de las opiniones usando indicadores potenciales de veracidad. Finalmente, usaron el algoritmo propuesto por Raykar *et al.* (2010) para asignar cada opinión del corpus a una clase.

Después de crear el corpus, los autores identificaron un conjunto de señales, de las cuales su presencia sugería que los textos eran engañosos. Las señales identificadas fueron las siguientes: 1) libros sospechosos, de acuerdo con la investigación, son libros que seguramente o muy probablemente recibieron opiniones engañosas; 2) el tiempo; Sandra Parker, una escritora de opiniones engañosas, relató que las agencias para las que trabajaba le daban máximo 48 horas para escribir las opiniones, por lo cual sugirió que se pusiera especial atención en libros que recibieran opiniones en cortos periodos; 3) registro, Amazon permite crear cuentas usando los nombres reales de los usuarios; por tanto, es menos probable que los que emplean su nombre real escribieran opiniones engañosas; 4) libro comprado, otra información que se puede obtener de Amazon es si el libro fue comprado, o no, mediante dicha página de internet; por lo tanto, la ausencia de compra del libro fue considerada una señal de engaño.

Fueron llevados a cabo dos experimentos: en el primero la asignación de clases se realizó con base en una votación de las señales de engaño. Aquellas opiniones con hasta 2 señales de engaño fueron consideradas veraces, y aquellas con más de 2 señales fueron consideradas engañosas. En el segundo fue usado el método de aprendizaje de multitudes.

Para realizar la clasificación se generó un vector de características por cada opinión. Dichas características se basaron en unigramas, bigramas y trigramas de lemas y de etiquetas gramaticales. Entonces, se generaron

dos listas de las frecuencias de cada lista de las fuentes de generación de características, una lista para opiniones engañosas y otra para opiniones veraces. Con ello, se buscaron las características más frecuentes de cada clase. Para esto, solamente los n-gramas que aparecieran más de 300 veces, en cada lista de frecuencias, fueron tomados en cuenta.

En la tabla III.9 se muestra la clasificación del conjunto de datos DeRev mediante una máquina de vectores de soporte. La tabla muestra también la comparación entre métodos de etiquetado, teniendo un mayor rendimiento, aunque no significativo, el uso del algoritmo de Raykar *et al.* (2010).

Tabla III.9: Clasificación del conjunto DeRev que muestra el rendimiento de los métodos de etiquetado

Método	Exactitud	Engañoso		
		P	R	F
Voto de clases	75.4	83.3	63.6	72.1
Algoritmo de Raykar *et al.* (2010)	76.3	78.7	72.0	75.2

Capítulo 4

Un nuevo enfoque para la detección de engaño

4.1. Fuentes de generación de características

En este estudio fueron investigadas específicamente tres fuentes diferentes de generación de características (LIWC, ns-gramas, MEP) destinadas a comparar y combinar el método propuesto (LDA).

La mayoría de las investigaciones, presentadas como las normas actuales (sección 3), utilizaron los unigramas como base para agregar nuevas características con el fin de obtener un mejor rendimiento. En cambio, para este estudio, se optó por utilizar un modelo de espacio de palabras (MEP) (sección 4.1.1), debido a que mostró un rendimiento similar, además de ser una representación más simple.

Dado que el engaño involucra procesos cognitivos, otra fuente de características que se seleccionó fue el diccionario LIWC (sección 4.1.3). Esta herramienta se ha utilizado para detectar el engaño generando información psicolingüística, debido a que fue creado para abordar estudios del área de psicología. LIWC consiste en un conjunto de palabras con relaciones semánticas.

Sin embargo, el principal inconveniente de LIWC es el conjunto de palabras preestablecidas. Por lo tanto, para poder comparar el desempeño de LIWC con respecto a una herramienta que también genera grupos de palabras con relaciones semánticas, se optó por utilizar un modelo de espacio semántico continuo (LDA) (sección 4.1.4). Este método crea automáticamente un conjunto de tópicos; cada uno consiste en un conjunto de palabras con ciertas relaciones semánticas.

A diferencia del LIWC, el conjunto de palabras de LDA puede cambiar dependiendo del conjunto de datos procesado, lo que sugiere que las categorías generadas son específicas de la colección de documentos, por lo que las características pueden ser más informativas.

Con los métodos descritos antes, sólo se ha considerado la información léxica, incluida la semántica léxica; sin embargo, no se incluyó información de estilo de texto. Por lo tanto, y dado que distintos trabajos muestran que la información sintáctica aporta información valiosa, se ha explorado un enfoque basado en el estilo de texto (ns-gramas) (sección 4.1.2). En contraste con otros métodos (BoW), los ns-gramas pueden considerar el contexto de las palabras en las oraciones

4.1.1. Matriz término-documento (MTD)

Los vectores de palabras se eligen de manera prominente en muchos trabajos debido al hecho de que generan características relevantes que ayudan a clasificar los textos. Por esta razón, se decidió analizar el rendimiento de las características basadas en una matriz de palabras, más específicamente, utilizando una representación binaria.

Para obtener una representación única, se forma una lista de todas las palabras $W_1, W_2, ..., W_n$ en el conjunto de datos. A continuación, se

analizó cada documento buscando si W_n existe en el texto actual; si ése es el caso, la característica n (F_n) se establece en 1, de lo contrario se establece en 0. En la figura IV.1 se muestra cómo los vectores de características están representados.

4.1.2. N-gramas sintácticos

Los n-gramas sintácticos (Sidorov *et al.*, 2014) (ns-gramas) son una característica relativamente nueva que surgió después de considerar ciertos inconvenientes de los n-gramas convencionales, siendo la principal desventaja de estos últimos que las relaciones de larga distancia no son correctamente capturadas, produciendo el efecto en el cual los n-gramas convencionales parecían ser generados de una manera demasiado aleatoria.

La principal ventaja de los ns-gramas es que como se extraen a partir de una estructura sintáctica, éstos pueden ser considerados como un fenómeno lingüístico. Esto no pasa con los n-gramas convencionales, generados de forma arbitraria.

La desventaja de poder generar n-gramas con información sintáctica es que se necesita un procesamiento sintáctico previo. Éste requiere de recursos adicionales como un analizador sintáctico para el lenguaje específico que se esté manejando; estos recursos no siempre están disponibles para todos los lenguajes.

De forma similar a los n-gramas convencionales, los ns-gramas pueden generarse a partir de diferentes tipos de elementos, por ejemplo, a partir de palabras, de etiquetas gramaticales o de relaciones sintácticas.

Para probar el rendimiento de los ns-gramas, en el estudio de Sidorov *et al.* (2014) fue clasificado un conjunto de datos de 39 documentos

escritos, en inglés, por tres autores diferentes. Asimismo, 60 % de los documentos fue usado como conjunto de entrenamiento y el 40 % restante fue usado como conjunto de prueba. Se muestra en la tabla IV.1 y la tabla IV.2 una comparación de rendimiento obtenida por los autores. En dichas tablas el tamaño del archivo determina el umbral de n-gramas considerados como los más frecuentes; la etiqueta NA significa que no se alcanzó el umbral establecido. Podemos apreciar que tanto para bigramas y trigramas sintácticos logran clasificar correctamente todos los documentos.

Con base en los buenos resultados obtenidos por los autores, se optó por usar los ns-gramas como fuente de generación de características para conocer su eficiencia en la detección de texto engañoso. Para tal propósito se detalla el procedimiento a continuación.

Para obtener ns-gramas, se construye un árbol sintáctico para recolectar relaciones sintácticas representadas por las aristas del árbol. Éstas son aquellas que unen palabras con la etiqueta apropiada.

Tabla IV.1: Comparación entre bigramas sintácticos y convencionales en la tarea de identificación de autoría

	Características			
Tamaño del archivo	ns-gramas de rel. sintácticas	n-gramas de POS tags	n-gramas de caracteres	n-gramas de palabras
400	100 %	90 %	90 %	86 %
1 000	100 %	95 %	95 %	86 %
4 000	100 %	NA	90 %	86 %
7 000	100 %	NA	NA	86 %
11 000	100 %	NA	NA	89 %

Tabla IV.2: Comparación entre trigramas sintácticos y convencionales en la tarea de identificación de autoría

Características				
Tamaño del archivo	ns-gramas de rel. sintácticas	n-gramas de POS tags	n-gramas de caracteres	n-gramas de palabras
400	100 %	90 %	76 %	81 %
1 000	100 %	90 %	86 %	71 %
4 000	100 %	100 %	95 %	95 %
7 000	100 %	100 %	90 %	90 %
11 000	100 %	95 %	100 %	90 %

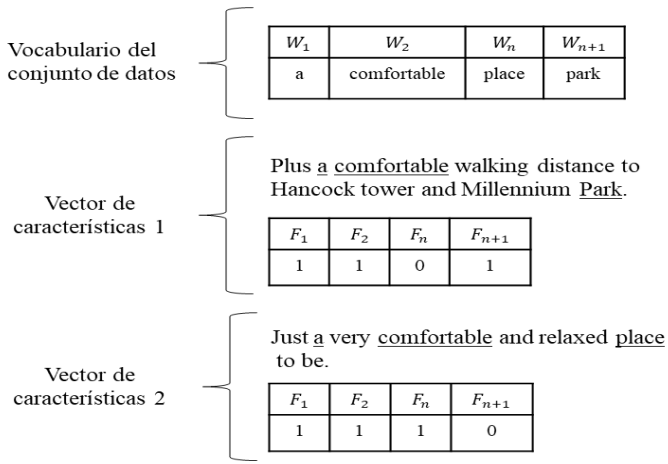


Figura IV.1: Ejemplo de construcción de vectores mediante un modelo de espacio de palabras

Se usó el analizador de Marneffe *et al.* (2006) para generar las relaciones sintácticas que corresponden a los documentos analizados. No se utilizó la representación colapsada. Se muestra un ejemplo de la representación de Stanford de las relaciones sintácticas en el paso 1 de la figura IV.2. En el paso 2, las relaciones útiles se seleccionan para generar

bigramas; se puede ver que prep_to (give-3, kid-7) está relacionado con det (kid-7, the-6) porque kid-7 los vincula a ambos. De esta manera, se obtiene el bigrama prep_to-det.

Los documentos pueden ser representados por sus relaciones sintácticas mediante el uso de un vector de características. Esto se hizo de la misma manera que se mostró en la sección anterior, ver figura IV.1, pero esta vez reemplazando las palabras por n-gramas sintácticos.

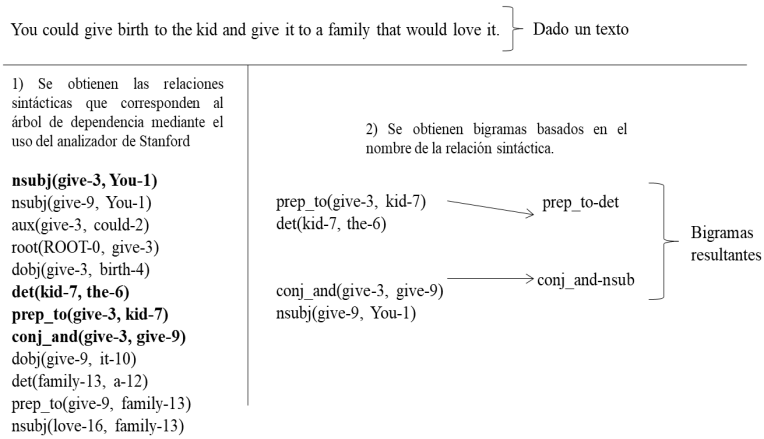


Figura IV.2: Proceso de generación de bigramas sintácticos dada una oración

4.1.3. Linguistic Inquiry and Word Count

Linguistic Inquiry and Word Count (LIWC) es una herramienta basada en un diccionario de palabras (Pennebaker *et al.*, 2015). El diccionario de LIWC contiene grupos de palabras etiquetados por

humanos y fue utilizado originalmente en trabajos relacionados con el análisis psicológico.

Las aplicaciones de este recurso han crecido recientemente; por ejemplo, el LIWC se ha utilizado en la lingüística computacional como una fuente de características para la identificación de autoría (Mohtasseb *et al.*, 2009) y detección de engaño (Masip *et al.*, 2012), entre otras.

En este trabajo se utilizó la versión 2007 de LIWC. Esta versión consta de cerca de 4 500 palabras agrupadas en 80 categorías o dimensiones. La figura IV.3 muestra un ejemplo de algunos grupos de palabras que podemos encontrar en LIWC.

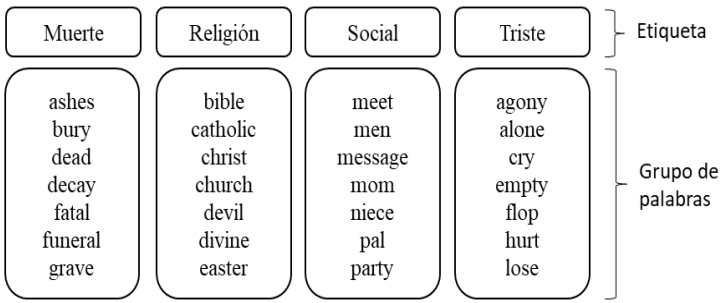


Figura IV.3: Ejemplo de algunos grupos de palabras, en LIWC, con su respectiva etiqueta. Es mostrado un grupo de sólo 7 palabras por etiqueta

4.1.4. Modelo de espacio semántico continuo-Latent Dirichlet Allocation (LDA)

Latent Dirchlet Allocation (LDA) es un modelo probabilístico generativo (Blei *et al.*, 2003) para colecciones de datos discretos, tales como colecciones de texto.

LDA representa a los documentos como una mezcla de diferentes tópicos; cada uno de éstos consiste en un conjunto de palabras que mantienen algún vínculo semántico entre ellas. Las palabras, a su vez, se eligen en función de una probabilidad. Se repite el proceso de selección de tópicos y palabras para generar un documento o un conjunto de documentos. Como resultado, cada documento generado se compone de diferentes tópicos.

De forma simplificada, el proceso generativo de LDA consta de los siguientes pasos:

1. La determinación de las N palabras que un documento contendrá de acuerdo con la distribución de Poisson.

2. La elección de una mezcla de tópicos para el documento de acuerdo con la distribución de Dirichlet con respecto a un conjunto fijo de K tópicos.

3. La generación de cada palabra en el documento mediante:

   - Elegir un tema
   - Utilizar el tema elegido para generar la palabra

Usando un modelo generativo de forma inversa, LDA analiza el conjunto de documentos para encontrar el conjunto más probable de tópicos posiblemente tratados en un documento.

Podemos considerar a LDA como una herramienta que genera grupos de palabras similares, tal como LIWC; pero a diferencia de este último, LDA genera automáticamente los grupos de palabras (tópicos). Además, los grupos de palabras de LDA no están etiquetados, y su contenido es

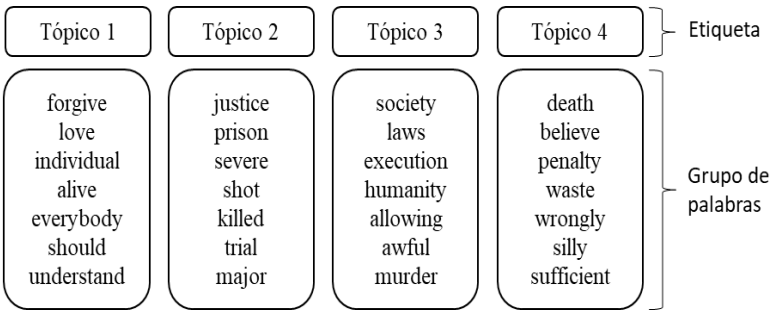


Figura IV.4: Ejemplo de generación de tópicos de textos del tema *pena de muerte*, mediante LDA

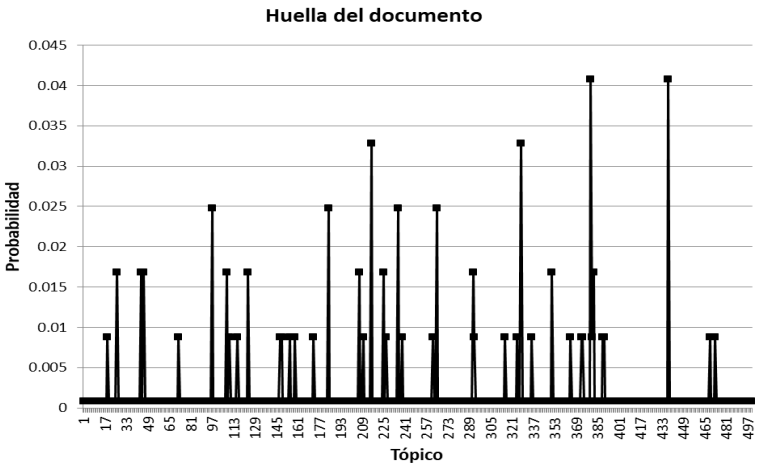


Figura IV.5: Ejemplo de documento procesado por LDA. Se muestra la distribución de tópicos

diferente dependiendo del corpus que sirve de entrenamiento. Vea la figura IV.4 para un ejemplo de palabras que se obtienen usando LDA.

La figura IV.5 muestra un documento procesado mediante LDA (antes de la binarización). Cada tópico mostrado tiene una probabilidad particular de existir dentro del documento actual. De esta manera, una

alta probabilidad de un tópico (x) en el documento (y) indica que es más probable que x aparezca en y. Cada documento de la colección tiene una distribución diferente de una cantidad fija de tópicos.

4.2. Conjuntos de datos

Para evaluar el método propuesto, fueron procesados tres conjuntos de datos diferentes: el corpus DeRev (DEception in REViews) (Fornaciari y Poesio, 2014), el corpus OpSpam (Opinion SPAM) (Ott *et al.*, 2011, 2013) y un corpus creado por opiniones sobre tres tópicos controversiales (Pérez Rosas y Mihalcea, 2014a).

El primer corpus fue recolectado utilizando el engaño sancionado, mientras que los dos últimos fueron recolectados usando un engaño no sancionado (Gokhman *et al.*, 2012). Los detalles sobre estos dos tipos de engaño se describen en las siguientes secciones.

4.2.1. Conjunto de datos DeRev

El conjunto de datos DeRev es un corpus compuesto de opiniones engañosas y veraces obtenidas de una página web de Amazon. Este corpus consiste en opiniones sobre libros. Es un estándar de oro (*gold standard*) que contiene 236 textos de los cuales 118 son veraces y 118, engañosos. Ambos casos fueron obtenidos de Amazon.

Para lograr un alto grado de confianza en una colección correctamente etiquetada de textos engañosos, los autores de DeRev consideraron dos investigaciones; la primera, de Sandra Parker, fue publicada en Money Talks News; mientras que la segunda, de David Streitfeld, apareció en *The New York Times*. Parker, por una parte, afirmó que recibió un pago

con el propósito de escribir opiniones sobre 22 libros. Streitfeld, por otra parte, dio a conocer cuatro libros en los que sus autores admitieron haber pagado por escribir opiniones engañosas; a partir de esto, se hizo un análisis y la selección se centró en 20 escritores de opiniones falsas, creando un corpus con 96 opiniones engañosas.

Finalmente, para obtener los 118 textos veraces, los autores del corpus DeRev tomaron en cuenta ciertos aspectos para asegurar una alta probabilidad de que la selección fuera correcta. La principal característica fue que los textos no deberían tener ninguna señal de engaño. Por esta razón, la selección se centró principalmente en aspectos tales como si las opiniones fueron escritas por usuarios que habían utilizado sus nombres reales, y si las opiniones fueron escritas por los usuarios que compraron el libro en Amazon, entre otros.

En este conjunto de datos, los textos engañosos y veraces no se obtuvieron de manera deliberada, es decir, a los participantes no se les pidió que escribieran engaños; en cambio, los textos se obtuvieron después de que el participante había mentido. Por esta razón, se dice que la forma de construir este texto es por engaño no sancionado. Se muestran a continuación un texto engañoso y veraz incluidos en éste:

Ejemplo de texto engañoso: *Circle of Lies, by Douglas Alan is a fast paced, gritty crime thriller that introduces us to John Delaney, ex-cop turned lawyer who finds himself defending his best friend against embezzlement and murder charges when no one else will believe him. Alan, a retired trial lawyer, gives us a rather unique look at the "men behind the curtains" of the American justice system with this fast paced, exciting read. Highly recommended!*

Ejemplo de texto veraz: *This book is well written and very funny, it is a good "guys"; kind of book, and it is an easy read, but don't let that fool you. The story is good and it is in the painful details. I felt like I was working right there with them. However, it is crude and not for the casual reader. I try to read both classic and non-classic writers of all eras, and I am always looking for something weird, funny, and cool to mix it up and this was it.*

4.2.2. Conjunto de datos OpSpam

El conjunto de datos OpSpam es un corpus compuesto de opiniones falsas y veraces. Éstas son opiniones sobre diferentes hoteles.

Para la generación de opiniones engañosas los creadores del corpus OpSpam utilizaron Amazon Mechanical Turk (AMT). Los autores mostraron a cada turker (trabajador de AMT) el nombre del hotel y su respectivo sitio web para que pudieran completar la tarea. Al utilizar el sitio web de AMT, se les pidió a los turkers lo siguiente: imaginaran que trabajaban en un hotel, el administrador les pidió que escribieran una opinión sobre el lugar, como si fueran invitados. La opinión debería parecer real y destacar los aspectos positivos del hotel. Las opiniones se limitaron a una por turker para evitar que la misma persona escribiera más de una. Asimismo, la tarea se limitaba a aquellos que vivían en los Estados Unidos; y, además, los turkers tendrían que alcanzar una calificación de aprobación de al menos 90 %.

Por una parte, los turkers tenían un máximo de 30 minutos para escribir la opinión y se les pagó un dólar por cada opinión aceptada. Con este procedimiento, los autores de OpSpam lograron recopilar 400 textos engañosos. Por otra parte, las opiniones veraces se recolectaron

de la página web de TripAdvisor. Por lo cual, 6 977 opiniones fueron extraídas de los 20 hoteles más populares. A continuación, los autores eliminaron 3 130 que no tenían cinco estrellas, 41 que no estaban escritas en inglés, 75 que tenían menos de 150 caracteres y 1 607 que habían sido publicadas por personas que opinaban por primera vez en Amazon. Finalmente, los autores de OpSpam seleccionaron 400 de los textos restantes.

Se recopilaron un conjunto de datos compuesto de 800 textos en total. Para formar este corpus, se pidió a los participantes que escribieran mentiras para obtener el texto engañoso. En consecuencia, el corpus se formó mediante el engaño sancionado. Se muestran a continuación un texto engañoso y veraz incluidos en éste:

Ejemplo de texto engañoso: *I had a wonderful time at the James Hotel while on business in Chicago. The rooms are modern, tasteful and well-kept, while the staff was responsive and efficient. This was the perfect place to unwind after working all day, but also provided an ideal atmosphere to do some more work.*

Ejemplo de texto veraz: *Despite what other are saying, this was one, if not the best Hotel stay in Chicago I have had. I travel to the Big City about three times a year for pleasure and The James rates up with the best. I was upgraded and the staff made the stay worth my special weekend visit. I don't believe you will be disappointed.*

4.2.3. Conjunto de datos de tópicos controversiales

Se trata de un corpus compuesto por 600 opiniones sobre tres temas controversiales: el *aborto* (200), la *pena de muerte* (200) y un *mejor amigo* (200). El corpus consta de 100 textos engañosos y 100 textos verídicos

por cada tópico (dando el total de 600 textos). La recopilación de textos se realizó a través de AMT y la tarea se limitó a los turkers que vivían en los Estados Unidos.

Para obtener textos veraces, los autores solicitaron a los participantes su opinión real sobre cada uno de los tópicos; después, se pidió a los participantes que mintieran acerca de su opinión real, mediante la cual se obtuvieron textos engañosos. El método utilizado para recolectar este corpus fue un engaño sancionado.

Dicho corpus también contiene textos en inglés de la India y en español de México; sin embargo, esos textos no fueron explorados en este trabajo.

Ejemplo de texto engañoso: *Abortion is despicable because we're giving women the choice to kill life. Where is the unborn child's voice? Do we not care about those who can't defend themselves? Abortion is systemic murder that needs to be stopped.*

Ejemplo de texto veraz: *Abortion is a choice. It should be made by both parties if possible, but the woman should have final say if there is disagreement. I think that with counseling, and all the other options available, it is not morally wrong for a woman to choose to have an abortion. A fetus is not viable outside the womb before 26 weeks, and then only barely, and with very expensive care.*

4.2.4. Análisis de los conjuntos de datos

Para proporcionar una visión más profunda de los corpora, se muestra en la tabla IV.3 un recuento de *tokens* y tipos de los diferentes conjuntos de datos descritos anteriormente, donde los *tokens* muestran la cantidad

total de palabras contenidas en los documentos y los tipos representan el número total de palabras no repetidas encontradas en documentos. Las palabras vacías (palabras que carecen de significado por sí solas) se mantuvieron para todos los experimentos.

Tabla IV.3: Cantidad de tipos y *tokens* en los corpora. PTD = Promedio de *Tokens* por Documento

Corpus	Número de docs.	*Tokens*	Tipos	PTD
OpSpam	800	96 793	6 469	121
DeRev	236	29 990	5 162	127
Aborto	200	15 958	1 997	80
Mejor amigo	200	11 717	1 718	59
Pena de muerte	200	15 615	2 034	78

Tabla IV.4: Promedio de tipos compartidos en los diferentes corpora

Corpus	Promedio de tipos compartidos		
	Engañoso+veraz	Engañoso	Veraz
OpSpam	19	19	19
DeRev	14	16	13
Aborto	14	12	16
Mejor amigo	9	8	12
Pena de muerte	14	13	15

La tabla IV.4 muestra el promedio de tipos con respecto a cada corpus. Esto es el promedio de palabras no repetidas que los documentos del corpus tienen en común. Los resultados se muestran tanto en todos los documentos (engañosos + veraces) como en clases separadas. Los resultados de esta tabla también indican que es menos probable que haya sesgos en los conjuntos, dado que el promedio de palabras compartidas es muy cercano entre clases unidas y clases separadas, lo

que sugiere que no hay demasiadas palabras predominantes por clase. Por el contrario, si éstas existieran, inferimos que el número de palabras compartidas debería haber disminuido significativamente, cuando las clases se combinaron, debido al hecho de que tendrían pocas palabras compartidas entre ellas.

4.3. Construcción de vectores de características

Se realizaron varios experimentos utilizando diferentes fuentes (descritas en la sección 4.1) y mezclas de éstas para encontrar la mejor combinación de características para la detección del engaño (de acuerdo con los conjuntos de datos analizados en este trabajo). Para todos los experimentos, la clasificación se llevó a cabo a través del algoritmo de aprendizaje automático Naïve Bayes Multinomial implementado bajo WEKA (Hall, 2009) con una validación cruzada de cinco pliegues.

En un intento de generar un modelo que represente mejor la detección de engaño, se implementó una selección de atributos con el propósito de eliminar características repetitivas e irrelevantes.

La selección de atributos (Guyon y Elisseeff, 2003) ayuda a obtener, en algunos casos, un modelo para identificar el texto engañoso con mayor eficiencia. También reduce las dimensiones de los vectores de características. Por ejemplo, en el presente estudio, los vectores compuestos de 4 000 características se redujeron a aproximadamente 60 características.

En el proceso de selección de atributos, la herramienta WEKA se usó con un evaluador basado en la correlación propuesta por Hall (1999); además, se utilizó un criterio de búsqueda basado en un algoritmo

de escalando la colina con retroceso (backtraking). Dicha combinación mostró un aumento significativo en la exactitud.

Se establecieron parámetros específicos para la selección de atributos en WEKA de la siguiente manera: el evaluador utilizado fue CfsSubSetEval (numThreads 1, poolSize 1) y la búsqueda de estrategia utilizada fue BestFirst (dirección: Forward, searchTermination 5).

Además, se han probado con otras estrategias de búsqueda como ExhaustiveSearch, pero la comprobación de todas las posibilidades en un gran conjunto de atributos fue muy demandante en términos de tiempo. Igualmente, se ha utilizado la búsqueda genética, la cual, aunque se desenvolvió más rápido, no proporcionó buenos resultados, incluso con el aumento del número de generaciones, tamaño de la población y el ajuste de la probabilidad de recombinación/mutación.

BestFirst, eventualmente, fue más eficiente aplicado a la representación de documentos utilizados en este trabajo. Un estudio más detallado de todos los enfoques de selección de atributos podría aportar mejores resultados.

Conjuntamente, se encontró que la binarización de los vectores de características dio lugar a un modelo más preciso para la detección de textos engañosos: La tabla IV.5 muestra los resultados de la detección de engaño con características LDA + MTD y el efecto de la aplicación de binarización y selección de atributos, la binarización se implementó antes del proceso de selección de éstos. Dado que usar convertir a valores binarios en conjunto con la selección de los atributos en todos los casos presentaron los mejores resultados, se usó esta configuración en todos los próximos experimentos. Las salidas detalladas (precisión, exahustividad

y medida F) de la binarización con selección de atributos (Binario con SeAt) indicadas en la tabla IV.5 se muestran en el capítulo 5.

A continuación, se detalla la conversión de todas las características en valores binarios.

- LIWC generó vectores de 64 características. Los medios para obtener cada vector fueron los siguientes: dado un documento y las 64 categorías, si una palabra de una categoría actual se encontró en el documento, entonces esa característica tenía el valor de uno, de lo contrario tenía el valor de cero.

- Los medios para generar vectores de características usando ns-gramas fueron los siguientes: primero fue formada una lista de ns-gramas no repetidos obtenidos en todos los documentos del conjunto de datos por analizar. A continuación, dado un documento y la lista de ns-gramas, si el ns-grama actual se encontró en el documento, entonces el valor de la característica se estableció en 1, de lo contrario se estableció en 0.

- Como se mencionó, LDA muestra, como resultado, vectores de características con valores reales (probabilidades de pertenecer a cada tópico). Por lo tanto, procedimos a convertir valores de las características en valores binarios. Para ello, se calculó un umbral dividiendo la suma de todas las probabilidades de pertenencia por el número de tópicos establecidos. Cada probabilidad que es igual o mayor que el umbral se convirtió en uno; de lo contrario se convirtió en cero.

Tabla IV.5: Comparación de exactitud (%) con respecto a los valores binarios y la selección de atributos (SeAt)

Corpus	Binarios+SeAt	No-binarios+SeAt	Binarios sin SeAt
Aborto	87.5	75.2	72.2
Mejor amigo	87.0	82.0	76.8
Pena de muerte	80.0	69.3	61.5
DeRev	94.9	74.9	88.8
OpSpam	90.9	88.5	87.2

4.4. Identificación del número de tópicos óptimo

LDA requiere que se especifique el número de tópicos que serán generados; cualquier cambio en este parámetro puede cambiar la exactitud de la clasificación. Por esta razón, es necesario encontrar un valor apropiado.

Tabla IV.6: Medida F obtenida con diferentes valores del número de tópicos

Tópicos	*Aborto*	*Mejor amigo*	*Pena de muerte*	DeRev	OpSpam
100	69.46	78.01	57.75	81.25	87.36
200	75.52	82.61	65.18	86.47	87.93
300	81.17	85.38	71.71	90.20	89.93
400	84.50	87.05	77.22	92.35	90.06
500	**86.70**	**87.58**	**80.41**	**94.53**	**90.10**
600	87.40	87.00	81.18	95.54	91.34
700	87.52	86.29	81.86	95.74	89.53
800	87.76	85.03	82.05	95.68	90.15
900	87.38	84.33	81.82	95.97	90.68
1 000	86.92	83.29	81.72	96.06	90.25

Para encontrar el número de tópicos que permitieron una clasificación óptima, se llevaron a cabo diversos experimentos. Los resultados de éstos se muestran en la tabla IV.6; ahí, el número de tópicos se comparan

con la medida F obtenida. También, se puede observar que al aumentar el número de tópicos es posible alcanzar un punto óptimo del cual el aumento del número de éstos no implica una disminución de la medida F (es decir, 500 tópicos).

Aunque es posible aumentar ligeramente la exactitud de algunos conjuntos de datos mediante el ajuste del número de tópicos, el interés fue encontrar un método general con características que producen un buen rendimiento con diferentes tipos de engaño. De esta manera, se buscó un conjunto de parámetros comunes que funcionen para todos los conjuntos de datos.

Dado lo anterior, en todos los experimentos se utilizaron 500 tópicos, es decir, cada documento procesado por LDA genera un vector de 500 características y cada uno de ellos está representado por una probabilidad de pertenecer a cada tema (sección 4.1.4, figura IV.5).

Capítulo 5

Evaluación del enfoque propuesto

5.1. Detección de engaño

En este capítulo se presentan los resultados detallados sobre la identificación del engaño con base en distintos métodos de generación de características.

Las tablas de cada corpus clasificado muestran los siguientes valores: exactitud, precisión (P), exhaustividad (R) y medida F (F).

Aunque la exactitud es una medida utilizada en muchas investigaciones sobre la detección de engaño y proporciona un punto de comparación con otros resultados, también se optó por mostrar precisión, exhaustividad y medida F; esto permite un análisis más profundo de los resultados. De esta manera, la precisión muestra el porcentaje de textos seleccionados que son correctos; mientras que la exhaustividad, el porcentaje de textos correctos que se seleccionan. Por último, la medida F es la medida combinada para evaluar la compensación P/R.

Se obtuvieron las medidas antes descritas para los siguientes métodos: latent Dirichlet allocation (LDA), un modelo de espacio de palabras (MTD: Matriz Término-Documento), n-gramas sintácticos (ns-gramas)

y Linguistic Inquiry and Word Count (LIWC); así como para las combinaciones de éstos.

Para probar la eficiencia de cada método de generación de características, su procesaron tres conjuntos de datos sobre diferentes temas: OpSpam, consiste en opiniones sobre hoteles; DeRev, se relaciona con opiniones sobre libros comprados en Amazon; y el conjunto de datos de tópicos controversiales, compuesto de opiniones sobre tres temas (*aborto*, *pena de muerte* y *mejor amigo*).

Con base en los conjuntos de datos disponibles, se investiga qué método es mejor en un único dominio 5.1.2, donde tanto el conjunto de entrenamiento y de prueba son del mismo tema, en un entorno de dominio mixto 5.1.3, donde tanto el conjunto de entrenamiento y de prueba consisten en una mezcla de tópicos, y en un entorno de dominio cruzado 5.1.4, donde los conjuntos de entrenamiento y prueba están formados por diferentes tópicos (sin intersección entre los conjuntos de prueba y entrenamiento).

Finalmente, se demostró que el enfoque propuesto para la detección automática de engaño puede ser usado en la tarea de identificación de autoría 5.3, obtendiendo resultados competitivos. Además, se comprobó que el método de generación de características es independiente del dominio y del lenguaje, dado que en esta segunda tarea se analizan textos de diferente idioma, dominio y género.

5.1.1. Medidas de validación

Al momento de proponer un método para la detección automática de texto engañoso, se requiere también de una medida para la validación

de los resultados. Es decir, una medida que entregue un valor númerico que tenga correlación con la eficiencia del método propuesto.

Una medida ampliamente aceptada por su robustez en el área de recuperación de información es la medida F; ésta indica la eficiencia de la clasificación. Asimismo, es también común usar el área bajo la curva, la cual indica en qué medida el clasificador tiene la capacidad de discriminar elementos de distinta clase.

Ambas brindan una forma objetiva de provar qué tan bien el método propuesto detecta texto engañoso sobre una colección.

5.1.1.1. Medida F

Por una parte, es una media armónica que comprende dos proporciones: exahustividad y precisión, alcanza su mejor valor en la unidad, donde se obtiene una precisión y una exahustividad perfecta. Asimismo, como se puede ver en la figura V.1, esta medida toma en cuenta distintos aspectos de una clasificación.

$$Precisión = \frac{VP}{VP + FP} \tag{5.1}$$

$$Exahustividad = \frac{VP}{VP + FN} \tag{5.2}$$

$$MedidaF = \frac{2 * Precisión * Exahustividad}{Precisión + Exahustividad} \tag{5.3}$$

Por otra parte, la precisión se centra en los documentos seleccionados (ecuación 5.1); mientras que la exahustividad toma

en cuenta todos los documentos relevantes (ecuación 5.2). De esta manera, la medida F considera los documentos bien seleccionados, pero también los documentos que se debieron de seleccionar y se discriminaron (ecuación 5.3).

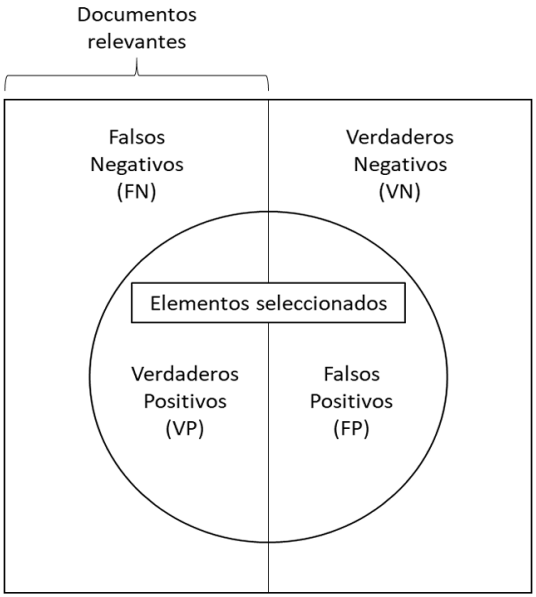


Figura V.1: Precisión y exhaustividad

5.1.1.2. Área bajo la curva ROC

La curva ROC (por sus siglas en inglés: Receiver Operating Characteristic) indica qué tan bueno es un clasificador binario para distinguir elementos de dos clases, por ejemplo si un texto es engañoso o veraz.

Esta medida involucra los aspectos de una tarea de clasificación mostrados en la figura V.1, y se basa en dos medidas: la tasa de

verdaderos positivos obtenida mediante la ecuación 5.4 (TPR, por sus siglas en inglés: True Positive Rate), y la tasa de falsos positivos obtenida mediante la ecuación 5.5 (FPR, por sus siglas en inglés: False Positive Rate). TPR y FPR son usadas para generar la curva ROC para finalmente medir el área bidemensional por debajo de la curva (AUC, por sus siglas en inglés: Area Under the Curve). El resultado del AUC es el indicador del rendimiento del clasificador.

$$TPR = \frac{VP}{VP + FN} \tag{5.4}$$

$$FPR = \frac{FP}{FP + VN} \tag{5.5}$$

5.1.2. Clasificación de un dominio específico

El objetivo de combinar diferentes fuentes de generación de características (sección 4.1) es encontrar cuál es la combinación que mejor favorece la exactitud y medida F en la clasificación de texto engañoso y veraz.

Las palabras contienen mucha información por sí mismas; esto puede ser confirmado por el gran número de investigaciones (sección 3) que utilizan unigramas como base para agregar nuevas características, esto con el objetivo de obtener un mejor rendimiento.

De manera similar, en este estudio se usó LDA para complementar las características de una MTD.

Los resultados en este capítulo muestran que las características generadas con la MTD en combinación con las de LDA obtienen mejores resultados que otros métodos utilizados (tales como, LIWC, ns-gramas) en la mayoría de los casos.

En la clasificación del conjunto de datos OpSpam, que se muestra en la tabla V.1, se puede observar que la combinación de LDA y MTD muestran un aumento en la exactitud con respecto a las mismas técnicas evaluadas por separado. Esto se debe a que las características generadas usando LDA complementan de una manera favorable las generadas por la MTD.

Existen casos en los que hay muchas características que representan el texto veraz, mientras que sólo algunas características representan texto engañoso o viceversa. LDA puede ayudar a minimizar este problema aumentando el número de características relevantes que conducen a una clasificación de texto más eficiente.

Para estudiar este efecto, fue enumerado el número de atributos obtenidos, después de implementar la técnica de selección de atributos, que están fuertemente vinculados con el texto engañoso o con el texto veraz, respectivamente. Por ejemplo, en 60 características relevantes, hay un número específico de características que están fuertemente correlacionadas con la clase engañosa, y otras características con la clase veraz (ver las dos primeras columnas de MTD en la tabla V.2).

Además, el número de características relevantes puede aumentarse combinando métodos de generación de características; en la tabla V.2 se muestra el número de características relevantes obtenidas para cada clase utilizando sólo la MTD en comparación con el número de características obtenidas mediante el uso de la combinación de LDA y MTD. Se puede ver que en todos los casos se presenta un aumento de características del texto engañoso.

Los vectores de características generadas por la combinación de características de LIWC y MTD, por un lado, mostraron resultados

similares, aunque con un rendimiento inferior en comparación con la combinación LDA + MTD. Sin embargo, la principal desventaja de LIWC, con respecto a LDA, es que la herramienta LIWC es etiquetada por humanos; esto significa que cada vez que se analiza un idioma diferente, es necesario obtener la herramienta para ese idioma específico. Por otro lado, LDA no requiere ninguna información dependiente del lenguaje para analizar un conjunto de datos.

Otro enfoque probado fue el uso de ns-gramas. Al igual que LIWC, es necesaria información previa dependiente del lenguaje, debido a que los árboles sintácticos se obtienen basándose en la probabilidad de ocurrencia en un corpus previamente etiquetado.

Los bigramas sintácticos mostraron una exactitud aceptable, pero la combinación de sus características puede empeorar los resultados. Por ejemplo, en los conjuntos de datos mostrados en la tabla V.3 y tabla V.5, la combinación de características entre ns-gramas y MTD disminuyó la exactitud de la clasificación. Por lo tanto, esta combinación específica de características era desventajosa.

Es importante señalar que con el método propuesto (LDA + MTD) se mantiene una exactitud competitiva para detectar textos engañosos en los diferentes conjuntos de datos clasificados, a diferencia de algunas otras técnicas del PLN, como la combinación entre relaciones sintácticas y unigramas utilizada por Feng *et al.* (2012).

El resultado de los autores es bueno en la evaluación del conjunto de datos OpSpam, que se muestra en la tabla V.1. Sin embargo, este desempeño no se mantuvo cuando los autores clasificaron conjuntos de datos sobre el *aborto*, el *mejor amigo*, y la *pena de muerte*; como se muestra en las tablas V.4-V.6, respectivamente.

Con base en el análisis de los resultados obtenidos, se encontró que la mejor combinación de características fue LDA y la matriz término-documento (LDA + MTD). Esta combinación muestra una exactitud del 90.9 % para el conjunto de datos OpSpam y del 94.9 % en el conjunto de datos DeRev. Se obtuvieron exactitudes del 87.5 %, 87.0 % y 80.0 % para los tópicos de *aborto*, *pena de muerte* y *mejor amigo*, respectivamente.

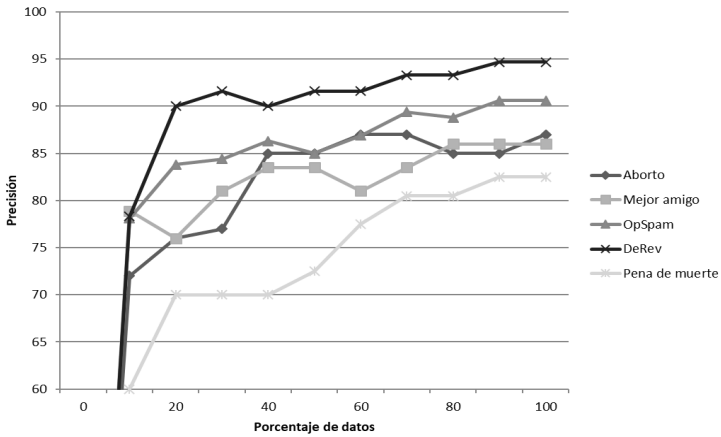


Figura V.2: Curva de aprendizaje de los diferentes corpora

La figura V.2 muestra el proceso de aprendizaje de cada conjunto de datos. Para cada porcentaje de datos muestreados al azar, del 10 % al 100 %, se utilizó una parte como conjunto de entrenamiento (80 %) y el resto (20 %) como prueba. Por ejemplo, si se utilizó el 60 % de los datos, digamos, 600 de 1 000 registros, 480 se utilizarían como entrenamiento y 120 como prueba. Este gráfico sugiere que, para algunos corpora, el uso de más datos podría producir un pequeño aumento en la exactitud,

mientras que para otros el aumento no es considerable entre 90 % o 100 % de todos los datos.

Tabla V.1: Clasificación del corpus OpSpam mediante el uso de diferentes fuentes de generación de características

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	72.0	72.8	70.3	71.5	71.3	73.8	72.5
MTD	87.9	87.0	89.0	88.0	88.7	86.8	87.7
ns-gramas (bigramas)	68.6	72.8	59.5	65.5	65.8	77.8	71.2
ns-gramas + MTD	86.0	88.7	82.5	85.5	83.6	89.5	86.5
LIWC	65.0	65.7	62.8	64.2	64.4	67.3	65.8
LIWC + MTD	87.9	89.4	86.0	87.6	86.5	89.8	88.1
LDA + MTD	90.9	94.8	86.5	90.5	87.6	95.3	91.3
LDA + LIWC	72.3	73.9	68.8	71.2	70.8	75.8	73.2
LDA + ns-gramas	74.9	72.6	80.0	76.1	77.7	69.8	73.5
LDA + MTD + LIWC	87.1	87.2	87.0	87.1	87.0	87.3	87.1
LDA + LIWC + ns-gramas	79.7	77.8	83.3	80.4	82.0	76.3	79.0
LDA + MTD + ns-gramas	88.9	88.6	89.3	88.9	89.2	88.5	88.8
LDA + MTD + LIWC + ns-gramas	88.1	88.2	88.0	88.1	88.0	88.3	88.1
Ott *et al.* (2011) (LIWC + bigramas)	89.8	89.8	89.8	89.8	89.8	89.8	89.8
Feng *et al.* (2012) (syntactic rel. + unigramas)	91.2	-	-	-	-	-	-
Fusilier *et al.* (2015)	90.2	-	-	-	-	-	-

5.1.3. Clasificación de dominios mezclados

El objetivo principal de los experimentos mostrados a continuación fue investigar el rendimiento en la clasificación cuando se combinan los conjuntos de datos de la tarea de detección de engaño de diferentes dominios. Con esto, el conjunto de entrenamiento contendría tópicos

Tabla V.2: Número de características relevantes obtenidas con la combinación de LDA y MTD

	MTD		MTD+LDA	
Corpus	Veraz	Engañoso	Veraz	Engañoso
Aborto	43	12	45	25
Mejor amigo	24	17	30	20
Pena de muerte	32	16	39	29
DeRev	27	52	37	57
OpSpam	31	22	31	24

Tabla V.3: Clasificación del corpus DeRev mediante el uso de diferentes fuentes de generación de características

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	80.5	79.0	83.1	81.0	82.1	78.0	80.0
MTD	91.9	90.9	93.2	92.1	93.0	90.7	91.8
ns-gramas (bigramas)	75.4	80.0	67.8	73.4	72.1	83.1	77.2
ns-gramas+MTD	91.1	89.4	93.2	91.3	92.9	89.0	90.9
LIWC	69.9	77.0	56.8	65.4	65.8	83.1	73.4
LIWC + MTD	91.5	92.2	90.7	91.5	90.8	92.4	91.6
LDA+MTD	94.9	94.2	95.8	95.0	95.7	94.1	94.9
LDA+LIWC	73.3	81.6	60.2	69.3	68.5	86.4	76.4
LDA+ns-gramas	78.4	83.2	71.2	76.7	74.8	85.6	79.8
LDA+MTD+LIWC	90.7	98.0	83.1	89.9	85.3	98.3	91.3
LDA+LIWC+ns-gramas	79.2	86.3	69.5	77.0	74.5	89.0	81.1
LDA+MTD+ns-gramas	91.5	90.2	93.2	91.7	93.0	89.8	91.4
LDA+MTD+LIWC+ns-gramas	83.0	84.8	80.5	82.6	81.5	85.6	83.5
Fornaciari y Poesio (2014)	76.27	-	-	-	-	-	-

que también incluiría el conjunto de prueba (pero no contendría textos de un sólo dominio).

Tabla V.4: Clasificación del tópico *aborto* mediante el uso de diferentes fuentes de generación de características

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	79.5	76.6	85.0	80.6	83.1	74.0	78.3
MTD	82.5	82.8	82.0	82.4	82.2	83.0	82.6
ns-gramas (bigramas)	65.8	73.2	52.0	60.8	62.8	81.0	70.7
ns-gramas+MTD	83.0	83.0	83.0	83.0	83.0	83.0	83.0
LIWC	72.5	72.7	72.0	72.4	72.3	73.0	72.6
LIWC + MTD	83.5	81.3	87.0	84.1	86.0	80.0	82.9
LDA+MTD	87.5	87.9	87.0	87.4	87.1	88.0	87.6
LDA+LIWC	80.0	76.8	86.0	81.1	84.1	74.0	78.7
LDA+ns-gramas	73.0	78.8	63.0	70.0	69.2	83.0	75.5
LDA+MTD+LIWC	76.5	76.2	77.0	76.6	76.8	76.0	76.4
LDA+LIWC+ns-gramas	77.0	76.5	78.0	77.2	77.6	76.0	76.8
LDA+MTD+ns-gramas	86.5	90.1	82.0	85.9	83.5	91.0	87.1
LDA+MTD+LIWC+ns-gramas	82.5	77.8	91.0	83.9	89.2	74.0	80.9
Pérez Rosas y Mihalcea (2014a)	80.3	-	-	-	-	-	-
Feng *et al.* (2012)(syntactic rel. + unigramas)	77.0	-	-	-	-	-	-

En este caso, nuevamente una combinación de características LDA y MTD produjo el mejor resultado (tabla V.7). Puede verse en la tabla V.8 que la SVM mostró un mejor rendimiento que el clasificador NB.

5.1.4. Clasificación de dominio cruzado

A diferencia de la clasificación que combina todos los dominios, para este experimento se seleccionó cada conjunto de datos como prueba y se combinaron los restantes como un conjunto combinado de entrenamiento. De esta manera, el dominio del conjunto de prueba no se incluyó en el de entrenamiento.

Tabla V.5: Clasificación del tópico *mejor amigo* mediante el uso de diferentes fuentes de generación de características

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	82.0	82.7	81.0	81.8	81.4	83.0	82.2
MTD	85.5	84.5	87.0	85.7	86.6	84.0	85.3
ns-gramas (bigramas)	73.0	81.1	60.0	69.0	68.3	86.0	76.1
ns-gramas+MTD	84.5	84.8	84.0	84.4	84.2	85.0	84.6
LIWC	75.5	74.3	78.0	76.1	76.8	73.0	74.9
LIWC + MTD	83.5	82.5	85.0	83.7	84.5	82.0	83.2
LDA+MTD	87.0	85.6	89.0	87.3	88.5	85.0	86.7
LDA+LIWC	78.0	73.7	87.0	79.8	84.1	69.0	75.8
LDA+ns-gramas	85.0	88.9	80.0	84.2	81.8	90.0	85.7
LDA+MTD+LIWC	77.0	72.5	87.0	79.1	83.8	67.0	74.4
LDA+LIWC+ns-gramas	79.5	78.6	81.0	79.8	80.4	78.0	79.2
LDA+MTD+ns-gramas	85.0	92.7	76.0	83.5	79.7	94.0	86.2
LDA+MTD+LIWC+ns-gramas	86.5	94.0	78.0	85.2	81.2	95.0	87.6
Pérez Rosas y Mihalcea (2014a)	75.9	-	-	-	-	-	-
Feng *et al.* (2012) (syntactic rel. + unigramas)	85.0	-	-	-	-	-	-

La tabla V.9 muestra los resultados de la clasificación interdominio. En estos experimentos, a diferencia de los presentados en las secciónes 5.1.2 y 5.1.3, la combinación de características de LDA y MTD no dio consistentemente la mejor exactitud.

En la mayoría de los casos (3 de 5), la SVM superó a NB; sin embargo, con una configuración relativamente simple de una matriz término-documento, NB es capaz de mejorar la detección de engaño con las características aprendidas de otros conjuntos de datos.

Tabla V.6: Clasificación del tópico *pena de muerte* mediante el uso de diferentes fuentes de generación de características

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	77.0	76.5	78.0	77.2	77.6	76.0	76.8
MTD	78.5	77.7	80.0	78.8	79.4	77.0	78.2
ns-gramas (bigramas)	61.5	66.2	47.0	55.0	58.9	76.0	66.4
ns-gramas+MTD	78.5	78.8	78.0	78.4	78.2	79.0	78.6
LIWC	65.5	64.5	69.0	66.7	66.7	62.0	64.2
LIWC + MTD	78.0	78.6	77.0	77.8	77.5	79.0	78.2
LDA+MTD	80.0	78.8	82.0	80.4	81.3	78.0	79.6
LDA+LIWC	66.0	79.6	43.0	55.8	61.0	89.0	72.4
LDA+ns-gramas	70.0	72.7	64.0	68.1	67.9	76.0	71.7
LDA+MTD+LIWC	61.0	60.4	64.0	62.1	61.7	58.0	59.8
LDA+LIWC+ns-gramas	75.0	78.4	69.0	73.4	72.3	81.0	76.4
LDA+MTD+ns-gramas	74.0	91.4	53.0	67.1	66.9	95.0	78.5
LDA+MTD+LIWC+ns-gramas	79.0	73.4	91.0	81.3	88.2	67.0	76.1
Pérez Rosas y Mihalcea (2014a)	77.2	-	-	-	-	-	-
Feng *et al.* (2012) (syntactic rel. + unigramas)	71.5	-	-	-	-	-	-

Tabla V.7: Exactitud, precisión (P), exhaustividad (R) y medida F (F) obtenidas con los corpora mezclados mediante SVM

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	65.8	67.6	60.9	64.1	64.4	70.8	67.4
MTD	73.5	73.6	73.5	73.5	73.5	73.6	73.5
LDA + MTD	76.3	76.8	75.6	76.2	75.9	77.1	76.5
LIWC	59.8	60.1	58.4	59.2	59.5	61.1	60.3
LIWC + MTD	73.7	71.9	77.8	74.7	75.8	69.7	72.6
LDA + LIWC	69.7	69.3	70.8	70.1	70.2	68.7	69.4
LIWC + MTD + LDA	72.6	72.3	73.5	72.9	73.0	71.9	72.5

Tabla V.8: Exactitud, precisión (P), exhaustividad (R) y medida F (F) obtenidas con los corpora mezclados mediante NB

		Veraz			Engañoso		
Método	Exactitud	P	R	F	P	R	F
LDA	61.1	61.5	59.3	60.4	60.7	62.8	61.7
MTD	74.3	73.9	75.3	74.6	74.8	73.3	74.1
LDA + MTD	73.5	71.9	77.0	74.4	75.3	69.9	72.5
LIWC	56.4	55.6	64.7	59.8	57.7	48.3	52.6
LIWC + MTD	73.3	73.3	73.3	73.3	73.3	73.3	73.3
LDA + LIWC	64.8	65.1	64.2	64.6	64.7	65.5	65.1
LIWC + MTD + LDA	73.3	76.0	68.3	71.9	71.2	78.4	74.6

5.1.5. Palabras y tópicos predominantes

Detectar engaño en un texto es una tarea que se ha tratado de mejorar mediante distintas técnicas. Dicha labor no es sencilla debido a que las características que distinguen a un texto engañoso pueden variar entre distintos conjuntos de datos.

Algunas palabras que representan el texto falso en un conjunto de datos pueden representar al texto veraz en otro; esto indica que una palabra no es estrictamente representante de una clase u otra en todos los casos. Por ejemplo, se puede observar en la tabla V.12 que el pronombre personal *I* y el adjetivo posesivo *my* son características representantes del texto veraz en el conjunto de datos *aborto*, en cambio, como se puede ver en la tabla V.10, estas dos palabras fungen como características representantes del texto engañoso en el conjunto de datos OpSpam.

El hecho de que ciertas palabras sean contradictorias en conjuntos de datos distintos no indica que los métodos automáticos de reconocimiento de texto engañoso sean disfuncionales, sino que estos vocablos pueden representar a un conjunto específico de datos, pero no figurarán al tener

Tabla V.9: Medida F obtenida en la clasificación de dominio cruzado. Se muestran resultados para los clasificadores NB y SVM

		DeRev	OpSpam	*Aborto*	*Mejor amigo*	*Pena de muerte*	Promedio
LDA	SVM	52.1	48.8	57.5	50.0	53.5	50.43
	NB	43.2	49.8	56.0	51.5	56.0	46.50
MTD	SVM	53.3	52.8	55.5	55.5	58.5	53.05
	NB	50.8	53.8	58.5	56.0	48.5	52.30
LIWC	SVM	51.3	49.2	53.1	54.3	52.8	50.25
	NB	50.9	51.1	52.6	56.8	51.4	51.00
LIWC + MTD	SVM	54.6	53.8	54.5	55.0	56.0	54.20
	NB	47.8	52.5	55.5	59.0	51.0	50.15
LDA + MTD	SVM	59.3	50.6	57.5	64.0	55.0	54.95
	NB	58.8	52.3	55.0	59.5	52.5	55.55
LDA + LIWC	SVM	56.3	46.3	54.5	55.5	52.0	51.30
	NB	45.7	48.1	46.5	51.5	49.5	46.90
LDA+LIWC+MTD	SVM	52.1	52.6	57.0	57.0	54.0	52.35
	NB	52.9	53.0	58.0	62.5	53.0	52.95
Mejor:		SVM	NB/SVM	NB	SVM	SVM	NB

un conjunto más general, por ejemplo, al unir el conjunto OpSpam con el conjunto *aborto*.

Se probó que la combinación de características LDA y MTD obtuvo una buena exactitud en los tres tipos de conjuntos de datos estudiados. Esto implica que los modelos generados mediante esta combinación de características son más fiables para la detección de texto engañoso, puesto que han sido probados en distintos casos de engaño y se han mantenido estables en cuanto a la exactitud obtenida.

Además, LDA+MTD puede ser aplicada a otros idiomas debido a que no requiere de algún corpus para genera las características como LIWC o n-gramas sintácticos.

Por una parte, el uso de LDA resultó ser en gran medida un método alternativo más eficiente que el uso de LIWC. Esto se debe a que LDA genera tópicos y toma en cuenta todas las palabras contenidas en los documentos, de tal forma que no habría palabras que escapen del proceso estadístico de generación de grupos de palabras. Por otra parte, LIWC contiene un grupo de palabras preestablecidas de las cuales probablemente algunas no estén incluidas en los documentos que se procesarán. Por lo tanto, la generación de un modelo que represente al texto engañoso puede ser más exitoso si se usa LDA.

Para mostrar cómo la combinación LDA y MTD se complementa favorablemente, en la tabla V.12 se muestra un ejemplo de las palabras y tópicos más relevantes del texto engañoso y del texto veraz en el conjunto de datos *aborto*. Un punto destacable de este ejemplo, por una parte, es que los tópicos son características dominantes en la detección de los textos engañosos, es decir, está compuesto de más tópicos que palabras únicas. Mientras que, por otra parte, las palabras son características dominantes en el texto veraz.

Se observó, en los conjuntos de datos analizados, que si una clase carece de características relevantes que acentúen una mayor diferencia entre clases, habrá tópicos que aumenten dichas características y mejoren la exactitud en la clasificación. En la tabla V.11, tabla V.13, y tabla V.14 se muestran los tópicos y palabras altamente discriminatorias entre la clase de texto engañoso y texto veraz. Los tópicos se pueden identificar

Tabla V.10: Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos OpSpam

Texto engañoso	Texto veraz
my	*location*
I	*floor*
luxury	*block*
relax	*bathroom*
spa	*street*
visit	*large*
vacation	*small*
anyone	2
amaze	*priceline*
chicago	*upgrade*

por ser un conjunto de palabras independientes, que no deben ser confundidas como una oración.

Tabla V.11: Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos *mejores amigos*

Texto engañoso	Texto veraz
he, is, guy, really, up, attention, overall, unintentionally, quite	*we*
this	*each*
his	*fried*
guy	*have*
lie	*same*
nice	*year*
put	*my*
trustworthy	*we, each, other, have, interests, but, over, one, stuff, others*
wonderful	*live*
how	*good*

Tabla V.12: Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz en el conjunto de datos *aborto*

Texto engañoso	Texto veraz
god	*I*
to, of, end, moment, cannot, give, consequences, society, perfect, return, couples, chance, that	*but*
is, not, this, womb, human, piece, so, there, inches, carry, conception, trash	*believe*
chance	*choice*
action	*or*
necessary	*me*
we, and, can, this, society, an, may, base, serve, contribute, nerves, started, effect, removed, his, always, subject	*for*
morally	*I, feel, life, me, know, even, people, bit, ends, experience, especially, unborn, control, sex, mother*
they, actions, believe, effectively, seriously, eyes, outcome, ask, mirror, ready	*my*
who, way, these, perform, themselves, where, example, available, always, out	*without*

5.1.6. Comparación de resultados y significancia estadística

La comparación de resultados es importante debido a que demuestra la efectividad del enfoque propuesto, además de mostrar las ventajas y desventajas con respecto a otros enfoques usados.

Tabla V.13: Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos *pena de muerte*

Texto engañoso	Texto veraz
instead-of	*I*
they, wouldn't, repetitive, capital, offenders, horrid, are, have, and, the	*believe*
him	*I, the, death, believe, penalty, that, am, waste, sufficient, wrongly, silly, because, be*
would, that, cases, also, upon, rape, used, get, someone, this, give, which, criminals	*put*
be, an, on, should, arm, everybody, individual, he, alive, year, forgive, concern, love, understand, please, had	*death*
it, people, justice, prison, first, severe, shot, trials, major, dictate, killed, trial	*for*
lesson	*opinion*
possibility	*sure*
the, because, humanely, promotes, per, population, future, offenders, equivalent, didn't, murderer, shows	*of*
people, as, murder, that, humankind, off, decide, alive, instead, undone, person	*current*

En la tabla V.15 se muestra una comparación entre el método propuesto y los resultados obtenidos por diferentes investigadores que

Tabla V.14: Los 10 tópicos y palabras más relevantes, representantes del texto engañoso y veraz del conjunto de datos DeRev

Texto engañoso	Texto veraz
author	*the, was, i, they, were, some, which, had, not, during, these, even, when, if, because, character, been, most, several*
your	*i, the, it, book, was, read, but, reading, first, all, my, had, this, that, one, so, characters, love*
uno/un	*history, american, native, our, americans, country, heart, indians, west, day, newspaper, the, tone, brown, better, known, fully, resist, stolen*
thriller	*classic*
the, and, of, a, to, is, in, that, this, for, book, as, it, on, with, an, are, but	*edition*
and	*I*
intrigue	*the, movie, book, than, seen, made, much, movies, version, like, cover, film, better, more, do, it, least, half, then*
guide	*version*
drug	*history*
issue	*again*

usaron los mismos conjuntos de datos con diferentes enfoques. Se puede observar que el enfoque propuesto en el presente estudio obtuvo un mejor

rendimiento en la clasificación de la mayoría de los conjuntos de datos, excepto para el conjunto de datos OpSpam.

Sin embargo, un mejor rendimiento o mayor exactitud no significa que exista una significancia estadística, a saber, si el incremento en la exactitud es significativo como para concluir que el enfoque es mejor o simplemente obtiene resultados equivalentes a otros métodos.

Por tal motivo, se muestra en la tabla V.16 la significancia estadística entre los resultados de esta investigación y los de otras investigaciones. Para fines de comparación, se establece un nivel de significancia (α) de 0.05 (5 %), lo que significa que la significancia estadística se alcanza si el valor p es menor que α. Con este nivel de significancia, el rendimiento mostrado por algunos de nuestros resultados no presentaría significancia estadística en comparación con otros métodos existentes, haciéndolos prácticamente equivalentes.

Tabla V.15: Comparación de nuestros resultados con otros estudios de los mismos corpora

Corpus	Estudios	Exactitud
OpSpam	Este estudio (LDA+MTD)	90.90 %
	Ott *et al.* (2011) (LIWC + bigramas)	89.80 %
	Feng *et al.* (2012) (syntactic rel.+ unigramas)	91.20 %
	Fusilier *et al.* (2015)	90.20 %
DeRev	Este estudio (LDA+MTD)	94.90 %
	Fornaciari y Poesio (2014)	76.27 %
Aborto	Este estudio (LDA+MTD)	87.50 %
	Pérez Rosas y Mihalcea (2014a)	80.30 %
Mejor amigo	Feng *et al.* (2012) (syntactic rel. + unigramas)	77.00 %
	Este estudio (LDA+MTD)	87.00 %
	Pérez Rosas y Mihalcea (2014a)	75.90 %
	Feng *et al.* (2012) (syntactic rel. + unigramas)	85.00 %
Pena de muerte	Este estudio (LDA+MTD)	80.00 %
	Pérez Rosas y Mihalcea (2014a)	77.20 %
	Feng *et al.* (2012) (syntactic rel. + unigramas)	71.50 %

Tabla V.16: Significancia estadística

Corpus	# docs.	Este estudio	Otros estudios	Valor-p	S. estadística
OpSpam	800	90.9 %	Ott *et al.* (2011) (89.9 %)	0.248	No
			Feng *et al.* (2012) (91.2 %)	0.416	No
			Fusilier *et al.* (2015) (90.2 %)	0.316	No
DeRev	236	94.9 %	Fornaciari y Poesio (2014) (76.3 %)	0	Sí
Aborto	200	87.5 %	Pérez Rosas y Mihalcea (2014a) (80.3 %)	0.025	Sí
			Feng *et al.* (2012) (77.0 %)	0.003	Sí
Mejor amigo	200	87.0 %	Pérez Rosas y Mihalcea (2014a) (75.9 %)	0.002	Sí
			Feng *et al.* (2012) (85.0 %)	0.283	No
Pena de muerte	200	80.0 %	Pérez Rosas y Mihalcea (2014a) (77.2 %)	0.248	No
			Feng *et al.* (2012) (71.5 %)	0.023	Sí

5.2. Impacto de la polaridad en los textos engañosos

Por lo general, las opiniones engañosas pueden ser positivas o negativas. Las primeras apuntan a persuadir a los compradores a elegir el producto; mientras las segundas intentan darle mala reputación.

Hasta el momento, hay algunos estudios que han tratado de descubrir señales universales de engaño (DePaulo *et al.*, 2003); sin embargo, no existe un conjunto universal de características que puedan identificar el engaño en el texto.

Esto sugiere que las señales de engaño podrían variar según la naturaleza de los textos. Por lo anterior, en esta sección se desarrollan experimentos para capturar características bajo el supuesto de que tomar en cuenta la polaridad del texto puede aumentar la exactitud de identificación de texto engañoso.

Las opiniones engañosas se han analizado en diferentes investigaciones, pero por lo común los autores muestran clasificaciones generales. Sin embargo, muchas características podrían descartarse debido a que las opiniones engañosas y veraces pueden ser positivas o negativas. Es decir, las opiniones negativas pueden incluir características particulares que pueden diferenciar sólo esa categoría de opiniones, por lo tanto, es posible que la polaridad de los textos afecta directamente las características que ayudan a identificar la verdad o el engaño.

Al examinar diferentes conjuntos de datos, se analiza el impacto de la polaridad en la clasificación del engaño en diferentes dominios. Se usaron los mismos conjuntos de datos disponibles para llevar a cabo las pruebas pertinentes y se realizaron diferentes experimentos con el objetivo de verificar en qué medida ciertas características son específicas de cada polaridad (positiva y negativa).

Asimismo, se implementó un clasificador de polaridad, como un paso de preprocesamiento, para llevar a cabo una clasificación específica. Los resultados muestran que hay señales específicas de engaño, las cuales pueden variar según la polaridad del texto. Además, existe una mejora

en la mayoría de los conjuntos de datos clasificados cuando se utiliza un clasificador de polaridad.

En los primeros estudios, la combinación de LDA + MTD arrojó los mejores resultados para la detección de engaño general. Sin embargo, debido a que las opiniones falsas suelen ser positivas o negativas, podría haber sido posible que la clasificación dependiera en gran medida de la polaridad de las opiniones.

Por ejemplo, un corpus podría consistir en 100 opiniones engañosas y 100 veraces; pero, a su vez, las primeras podrían ser en su mayoría positivas y las segundas podrían ser en su mayoría negativas. En consecuencia, fue considerada la suposición de que el clasificador podría tender a identificar características correspondientes a la polaridad de los textos, en lugar de las que corresponden al engaño en los textos.

Para descartar la suposición considerada, se llevó a cabo un experimento simple. Suponiendo que la exactitud de la clasificación debería disminuir en el caso de que la presencia de palabras de polaridad estuviera vinculada con la clasificación de opinión, se excluyeron palabras positivas y negativas utilizando un lexicón de polaridad (Hu y Liu, 2004).

Éste se creó mediante el procesamiento de un conjunto de opiniones sobre ciertos productos, y está en constante crecimiento. Actualmente, consta de 2 006 palabras positivas y 4 783 negativas seleccionadas de opiniones subjetivas. Los autores seleccionaron opiniones subjetivas y se basaron en el hecho de que la presencia de adjetivos puede determinar una expresión de opinión; además, limitaron la selección a aquellas opiniones que se formaron a partir de oraciones relacionadas con la calidad de los productos.

Como puede verse en la tabla V.17, la medida F obtenida después de excluir las palabras de polaridad apenas disminuyó con respecto a la medida F obtenida con los corpora originales. Las palabras en sí mismas podrían ser sólo un indicador de la polaridad del texto, pero no un factor determinante para identificar engaño. Por lo tanto, estos resultados muestran que la clasificación no depende de las palabras positivas y negativas como características importantes.

Por una parte, el hecho de que las palabras positivas y negativas no afecten los resultados indica que agregar la polaridad como característica a los vectores no es la mejor manera de mejorar la detección de engaño.

Por otra parte, con base en el hecho de que existe alguna evidencia que las características del engaño pueden variar según la naturaleza del texto (Hernández Castañeda y Calvo *et al.*, 2017), se esperó mejorar la detección del engaño realizando una clasificación de polaridad antes de capturar las características.

Tabla V.17: Resultados de la clasificación excluyendo palabras de polaridad

Corpus	P	R	Medida F	Medida F, corpus sin cambios
Aborto	0.871	0.870	0.870	0.875
Mejor amigo	0.87	0.870	0.870	0.870
Pena de muerte	0.819	0.815	0.814	0.800
OpSpam	0.865	0.865	0.865	0.867
DeRev	0.945	0.945	0.945	0.949

Tabla V.18: Clasificación del corpus OpSpam, etiquetado por los autores (OAP) y el etiquetado realizado en este estudio (PTW), basado en la polaridad del texto

Polaridad	# docs.	P	R	F
POSITIVOS-OAP	800	0.912	0.909	0.910
NEGATIVOS-OAP	800	0.844	0.844	0.844
POSITIVOS-PTW	620	0.828	0.826	0.826
NEGATIVOS-PTW	980	0.843	0.843	0.843
POS+NEG	1600	0.867	0.867	0.867

Tabla V.19: Clasificación de los conjuntos de datos agregando característica de polaridad

Corpus	P	R	F
Aborto	0.871	0.870	0.870
Mejor amigo	0.865	0.865	0.865
Pena de muerte	0.815	0.810	0.809
DeRev	0.928	0.928	0.928
OpSpam	0.865	0.865	0.865

5.2.1. Clasificación de polaridad para mejorar la detección de engaño

Los resultados de la sección 5.2 podrían sugerir que el clasificador no toma en cuenta las características relacionadas con la polaridad. Sin embargo, como se ha dicho, hay alguna evidencia que indica que la polaridad puede ser útil para identificar el engaño en el texto. Por lo tanto, aunque las palabras de polaridad no influyen directamente en la clasificación del texto engañoso, tales palabras pueden ayudar indirectamente al clasificar previamente las opiniones positivas

Tabla V.20: Comparación de la medida F obtenida antes (F-Ant) y después (Avg. F) de la clasificación de polaridad. Las medidas precisión (P), Exahustividad (R), medida F (F) y el número de opiniones por clase (# docs.) son también mostrados

Corpus	Positivas				Negativas					
	docs.	P	R	F	docs.	P	R	F	Avg-F	F-Ant
Aborto	49	0.931	0.918	0.919	151	0.895	0.894	0.894	0.906	0.875
Mejor amigo	121	0.868	0.868	0.868	79	0.863	0.861	0.861	0.864	0.870
Pena de muerte	87	0.885	0.885	0.885	113	0.868	0.867	0.867	0.867	0.800
DeRev	158	0.930	0.930	0.930	78	0.987	0.987	0.987	0.958	0.949
OpSpam	620	0.826	0.826	0.826	980	0.843	0.843	0.843	0.835	0.867

y negativas (como un paso de preprocesamiento) y luego clasificar las opiniones como veraces o engañosas.

En este punto, las características generadas serían específicas para cada clase de polaridad. Por esta razón, se realizó una clasificación parcial en lugar de una clasificación general. Es decir, las opiniones se separan primero según su polaridad para mejorar los resultados de clasificación.

Como primera evaluación de esta estrategia, se optó por utilizar el corpus OpSpam dado que los autores ya lo han dividido en opiniones positivas y negativas (Ott *et al.*, 2013).

La clasificación basada en la polaridad del corpus OpSpam, mostrada en la tabla V.18, consistió en clasificar opiniones positivas, negativas y la combinación de éstas. En este punto, hay dos métodos para dividir el corpus de OpSpam por polaridad: la clasificación de los autores OpSpam (OAP) (Ott *et al.*, 2013) y la clasificación realizada en este trabajo (PTW). La primera, por ser clasificada por los autores, se considera como un estándar de oro. Los registros de OAP muestran los resultados de la

clasificación teniendo en cuenta la división de los autores de OpSpam, mientras que los registros de PTW muestran los resultados con el corpus dividido por un clasificador de polaridad desarrollado en este estudio.

Con el objetivo de construir un clasificador supervisado de opiniones positivas y negativas, seguimos el trabajo de Pang *et al.* (2002); así se obtuvieron unigramas de un corpus que contiene reseñas sobre películas. Luego, se ordenaron por frecuencia para hacer un diccionario de 16 165 elementos. Para formar el conjunto de entrenamiento, se leyó cada opinión de las películas para hacer un vector que consiste en frecuencias de palabras, es decir, si la palabra existe, se aumenta el conteo de esa característica. Finalmente, se usó un clasificador bayesiano para procesar vectores. Este método basado en el aprendizaje automático sólo incluía información léxica; sin embargo, ha demostrado obtener resultados aceptables. Explorar con otros enfoques para la clasificación de la polaridad podría mejorar la eficiencia en la detección de engaño.

Los resultados muestran que el clasificador de polaridad podría no haber obtenido una clasificación óptima, ya que el promedio de la medida F de PTW es menor que OAP. Por lo tanto, si este comportamiento se mantiene, los resultados de las clasificaciones para los corpora restantes podrían mejorarse utilizando un clasificador de polaridad más robusto.

Fue aplicado este clasificador a los conjuntos de datos restantes, de modo que ahora cada uno de ellos está dividido en opiniones positivas y negativas (y cada clase tiene sus opiniones verdaderas y engañosas, respectivamente). La tabla V.20 muestra la cantidad de opiniones positivas y negativas para cada corpus. Con este experimento se pone a prueba el método propuesto y se comprueba que existen ciertas mejoras en la clasificación.

Para comparar la efectividad entre hacer una clasificación previa de polaridad y simplemente agregar características de polaridad a los vectores, se realizó un segundo experimento donde se agrega información de polaridad como una característica. Ésta se incluyó en los patrones antes de proceder a clasificar los corpora. La diferencia radica en el hecho de que, en este caso, todo el corpus se toma en cuenta y se agrega una característica más que indica si la opinión es positiva o negativa.

Para determinar la polaridad, el clasificador genera un valor real entre 0 y 1; por lo tanto, el valor de la característica se establecerá en 1 si el valor real está por encima de 0.5, de lo contrario se establecerá en 0.

Como se puede ver en la tabla V.19, por un lado, la medida F apenas cambia, en el caso de tópicos controvertidos (*aborto*, *mejor amigo* y *pena de muerte*). Por otro lado, la clasificación DeRev y OpSpam mostró que la medida F se redujo en mayor medida. Estos resultados indican que incluir información de polaridad en los patrones (como característica adicional) no es útil e incluso es perjudicial para la detección de engaño.

Todo lo anterior demuestra que obtener características específicas al dividir el texto en positivo y negativo antes de buscar señales de engaño (enfoque propuesto para este trabajo, consulte tabla V.20) mostró mejores resultados que simplemente agregar información de polaridad como característica adicional (tabla V.19).

5.3. Resolución de una tarea diferente

Debido a que el enfoque propuesto es independiente del dominio y del lenguaje, es posible usarlo para otras tareas desde que los algoritmos de

aprendizaje automático pueden generar modelos basados en el conjunto elegido de datos.

En esta sección, se muestra como LDA puede ser aplicado a diferentes tareas de clasificación.

5.3.1. Identificación de autoría

La verificación de autoría es un problema importante que resolver, debido a que muchas tareas requieren el reconocimiento del autor que escribió un texto específico. Por ejemplo, saber qué autor escribió un libro anónimo o identificar notas de un asesino en serie.

En esta sección, se aborda el desafío de verificación del autor en un entorno más realista. Los conjuntos de datos utilizados (2 de ellos) consisten de 1 a 5 documentos de un autor conocido y un documento de un autor desconocido. Los conjuntos de datos están formados por subconjuntos en diferentes idiomas: inglés, español, holandés y griego. El objetivo es identificar si un texto desconocido fue escrito por el mismo autor que escribió los textos conocidos.

Es importante tener en cuenta que esta tarea se vuelve más difícil cuando el conjunto de datos se compone de documentos cortos, ya que los enfoques comunes no pueden capturar modelos efectivos con pocas cantidades de palabras (Stamatatos, 2009). Sin embargo, en casos reales dentro del campo forense, los textos largos rara vez existen.

Se han propuesto diversos enfoques para generar características más informativas basadas en el estilo del texto. También es posible generar características extrayendo información léxica, sintáctica o semántica, entre otras.

La información léxica se suele limitar a los recuentos de palabras y la aparición de palabras comunes. La información sintáctica es capaz de obtener, en cierta medida, el contexto de las palabras.

En este trabajo se usó información semántica para encontrar características que ayuden a discriminar textos. Para este propósito, fue creado un modelo utilizando LDA. Al usar este método, se tomó en cuenta el vocabulario de todos los textos al mismo tiempo, y después de un proceso estadístico, fue descubierto hasta qué punto se dan las relaciones entre las palabras en cada documento.

Se evaluó el enfoque propuesto en dos conjuntos de datos en los que se aborda una tarea de verificación del autor: el corpus PAN 2014 y 2015.

5.3.2. Enfoques para la verificación de autoría

Diversos trabajos han intentado estudiar el desafío de identificación de autor mediante la generación de diferentes tipos de características (Madigan *et al.*, 2005; Narayanan *et al.*, 2012).

La naturaleza del conjunto de datos puede determinar la dificultad de la tarea, es decir, qué tan difícil será extraer las características apropiadas (Afroz *et al.*, 2005; Green *et al.*, 2013). En Nirkhi y Dharaskar (2013), se puede encontrar un estudio de técnicas, que muestra que, si bien el número de autores aumenta y el tamaño del conjunto de datos de entrenamiento disminuye, el rendimiento de la clasificación también disminuye. Esto parece lógico, debido a que el tamaño de los datos de entrenamiento es menor y, por lo tanto, la identificación de características útiles se ve afectada.

Muchos estudios analizan la identificación del autor a través de su estilo de escritura (Pateriya *et al.*, 2012; Pavelec *et al.*, 2007; Verhoeven y Daelemans, 2014).

Por ejemplo, en Layton *et al.* (2013), las características basadas en estilo se comparan con el método BoW (Bolsa de palabras). Se intenta discriminar a los autores de los textos en el mismo dominio obtenidos de Twitter. Para generar los vectores de características se incluyen marcadores de estilo como caracteres, palabras largas, espacios en blanco, puntuación, hipervínculos o partes del habla, entre otros. Los autores encontraron que un enfoque basado en el estilo era más informativo que un método basado en BoW; sin embargo, sus mejores resultados se obtuvieron al considerar dos autores, por lo que hubo una disminución en la exactitud cuando se incrementó el número de autores. Esto sugiere que, dependiendo de qué tan grande es el conjunto de entrenamiento, habrá características estilísticas que ayudan a distinguir a un autor de otro, pero no a cualquier otro autor.

Las características estilísticas también se pueden aplicar a otras tareas. En Bergsma *et al.* (2012) se combinan características para tratar problemas de dos clases. Intentaron obtener características de estilo, BoW y sintácticas para clasificar a los escritores nativos y no nativos de lengua inglesa, textos escritos para conferencias o talleres (clasificación del lugar) y textos escritos por escritores masculinos o femeninos (clasificación de género). El conjunto de datos consta de textos que son artículos científicos; este tipo de escritos es más extenso, a diferencia del correo electrónico, los tuits y otros textos breves. Por lo tanto, se puede suponer que esto podría haber llevado a identificar textos escritos por no nativos con una exactitud prometedora. Sin embargo,

los textos largos no garantizan buenos resultados, ya que sus tareas de clasificación por lugar y género obtuvieron un bajo rendimiento.

El propósito de identificar la autoría puede variar. Por ejemplo, Bradley *et al.* (2008) intentaron probar que es posible averiguar qué autor escribió un artículo no publicado (para una conferencia o un diario); sólo considerando las obras citadas en éste. Al utilizar el análisis semántico latente (LSA, por sus siglas en inglés: Latent Semantic Analysis) Dumais (2004) propone crear una matriz de términos y documentos en la que los posibles autores se consideran documentos, mientras que los autores citados se consideran términos. LSA asume que las palabras que tienen un significado cercano aparecerán en partes de texto similares. Los resultados de Bradley *et al.* (2008) demostraron que el sistema de revisión a ciegas debería ser cuestionado.

Otro ejemplo es el trabajo de Castro y Lindauer (2012), con la tarea de averiguar si la identidad de los usuarios de Twitter puede ser descubierta por su estilo de escritura. Los autores se enfocaron en características como la forma y longitud de la palabra, las frecuencias de los caracteres y de las palabras vacías, entre otras. Con un algoritmo RLSC (por sus siglas en inglés, Regularized Least Square Classification), los autores clasificaron correctamente 41 % de los tweets.

En Pimas *et al.* (2015), la tarea de verificación del autor se aborda mediante la generación de tres tipos de características: estilométricas, gramaticales y estadísticas. El estudio de dichos autores se basa en el desafío de verificación de autoría de PAN 2015. La distribución de tópicos también se considera, pero argumentan en contra de su uso porque el conjunto de datos está formado por mezclas de tópicos de una manera que afectó sus resultados. Por un lado, un modelo de validación cruzada

(10 pliegues) muestra un buen rendimiento; por otro lado, el modelo mostró un sobre entrenamiento utilizando los conjuntos de prueba y entrenamiento especificados.

5.3.3. Conjuntos de datos

El enfoque propuesto fue probado en dos conjuntos de datos, los cuales son incluidos en PAN 2015, que se basa en documentos de tema cruzado y género cruzado, y PAN 2014, que consiste en documentos de dominio específico.

Tabla V.21: Detalles del conjunto de datos para la tarea PAN 2015 de identificación del autor

	Entrenamiento			Prueba			
Lenguaje	Problemas	# docs.	Prom. palabras × doc.	Problemas	# docs.	Prom. palabras × doc.	Género
Inglés	100	200	366	500	452	536	Mezcla-tópico
Español	100	500	954	100	1000	946	Mezcla-tópico/género
Holandés	100	276	354	165	380	360	Mezcla-género
Griego	100	100	678	100	500	756	Mezcla-tópico

Tabla V.22: Detalles del conjunto de datos para la tarea PAN 2014 de identificación del autor

	Entrenamiento			Prueba			
Lenguaje	Problemas	# docs.	Prom. palabras × doc.	Problemas	# docs.	Prom. palabras × doc.	Género
Holandés	96	268	412.4	96	287	398.1	Ensayos
Holandés	100	202	112.3	100	202	116.3	Revisiones
Inglés	200	729	848.0	200	718	833.2	Ensayos
Inglés	100	200	3 137.8	200	400	6 104.0	Novelas
Griego	100	385	1 404.0	100	368	1 536.6	Artículos
Español	100	600	1 135.6	100	600	1 121.4	Artículos

5.3.3.1. Conjunto de datos PAN 2015

Para realizar experimentos con nuestro enfoque, fue usado el corpus propuesto en la tarea de identificación del autor de PAN 2015 (Stamatatos *et al.*, 2015).

Este conjunto de datos consta de cuatro subconjuntos, cada uno escrito en diferentes idiomas: inglés, español, holandés y griego. Los subconjuntos tienen diferencias significativas.

El subconjunto de inglés consiste en líneas de diálogo de jugadas; el de español reside en artículos de opinión de periódicos en línea, revistas, blogs y ensayos literarios; el holandés está formado por ensayos y revisiones y el subconjunto griego está formado por artículos de opinión de categorías como política, salud, deportes, entre otros.

El corpus también tiene una cantidad diferente de documentos por subconjunto, como se detalla en la tabla V.21. Además, cada idioma consta de una serie de dificultades para resolver que se definen específicamente a continuación (sección 5.3.4).

Debido a su naturaleza, este conjunto de datos se centró en problemas que requieren la captura de información más específica sobre la forma de escribir del autor. Por ejemplo, supongamos que conocemos a una persona que trabajó para un periódico escribiendo artículos sobre deportes; sin embargo, dicho sujeto decide ser independiente y pasa la vida escribiendo novelas de terror. Una posible tarea, por un lado, podría ser averiguar qué artículos pertenecen al exescritor deportivo entre artículos deportivos de diferentes autores. En este caso, el vocabulario de los documentos puede descubrir al autor; por ejemplo, por su tasa de uso de n-gramas, este tipo de tarea se le llama de tema cruzado.

Por otro lado, otra tarea posible es descubrir si una novela de terror fue escrita por el novelista, basándose en los artículos deportivos que él o ella escribieron antes. En consecuencia, hay un cambio drástico en el género y el tema de los documentos, es decir, la intersección entre los vocabularios de los documentos se reduciría sustancialmente. Debido a esto se le llama tarea de género cruzado.

5.3.3.2. Conjunto de datos PAN 2014

El corpus PAN 2014 (Stamatatos *et al.*, 2014), al igual que PAN 2015, consta de documentos escritos en cuatro idiomas: holandés, inglés, griego y español. Sin embargo, los documentos de PAN 2014 están en el mismo dominio, es decir, no contienen mezclas de género.

Cada conjunto de datos contiene uno o hasta cinco documentos conocidos y uno desconocido; el desafío es averiguar si este último fue escrito por el mismo autor que escribió los documentos conocidos. Se muestran en la tabla V.22 detalles sobre el corpus PAN 2014.

5.3.4. Método propuesto

Como se mostró en las sección 5.3.2, las características basadas en estilística, información sintáctica y léxica consideran cada documento escrito por separado. Por ejemplo, es posible obtener características estilísticas contando el número de palabras vacías y caracteres por documento. Estas apariciones de símbolos, en textos cortos y de dominios cruzados, pueden ser aleatorios y no permiten capturar el estilo de escritura. En una colección de textos de dominios cruzados, inferimos que las palabras vinculadas y su distribución en los textos pueden proporcionar características más informativas, debido a que la

distribución de tópicos estimada de LDA dependerá del contenido de cada texto de la colección.

5.3.4.1. Método para la verificación de autoría

Existen diferentes investigaciones que han utilizado LDA como fuente de características. Por ejemplo, Pacheco *et al.* (2015) se enfrentaron al desafío de verificación del autor de PAN 2015 al proponer un esquema basado en el modelo de fondo universal (Reynolds *et al.*, 2000). Mediante la aplicación de ese esquema crearon tres vectores de características: un vector para el autor, otro para cada conjunto de documentos conocidos y uno más para cada documento desconocido. Los tres vectores estuvieron codificados para producir uno nuevo. Luego, las características (incluidas las generadas por LDA) se seleccionaron mediante un bosque aleatorio que determina jerárquicamente la importancia de cada característica.

Del mismo modo, Moreau *et al.* (2015) usaron un algoritmo genético para seleccionar las mejores fuentes de características (estrategias) para la verificación del autor de PAN 2015. Una de las estrategias fue LDA con la cual se generaron dos vectores de cinco tópicos al ejecutar los siguientes pasos: primero, los autores se dividieron documentos en caracteres n-gramas; segundo, se indagó un conjunto de cinco tópicos para el texto de cada autor; por último, se buscó un conjunto de cinco tópicos para verificar cada documento (documento desconocido). Con los dos vectores, los autores obtuvieron una medida que indica qué tan similares son los vectores, con respecto al estilo; esa medida es una característica del algoritmo genético.

Savoy (2013), de manera similar al enfoque que propone el presente libro, utiliza la distribución de tópicos calculada por LDA. El desafío

es averiguar qué autor, de un conjunto de ellos, escribió un documento específico. Fue calculada la distancia simétrica de Kullback-Leibler entre la distribución del documento desconocido y la de los documentos de los autores. Entonces, el autor se asignó con respecto a la distancia más cercana. Esta tarea se denomina atribución de autoría debido a que el desafío es identificar a los autores de texto anónimo.

La tarea de atribución de autoría también fue estudiada por Seroussi *et al.* (2014); en sus temas de investigación, las distribuciones se vieron como vectores (patrones) que se clasificaron directamente.

La principal diferencia entre los otros trabajos mencionados anteriormente es que, en éste, se generaron características mediante el uso de LDA bajo el supuesto de que cierto autor tiene una forma específica de relacionar palabras y, por lo tanto, genera un texto. Esa suposición condujo a analizar qué palabras mantienen algún enlace en un texto escrito y la medida en que el conjunto de palabras vinculadas están presentes en el texto. Con esta información, producimos restas de vectores para proporcionar nuevos vectores que forman un conjunto de entrenamiento; este proceso se detalla a continuación:

Proponemos utilizar, en específico, LDA para extraer información semántica del corpus. Como se mencionó anteriormente, por un lado, dada una colección de textos, LDA puede encontrar relaciones entre las palabras según la forma en que se usan en el texto. Por otro lado, los enfoques de estilística comunes utilizan tasas de símbolos en los textos para distinguir dos documentos escritos por autores diferentes. Como se ha dicho anteriormente (sección 5.3.2), mientras que los textos se vuelven más cortos, la cantidad de símbolos tiende a no ser suficiente

para producir características discriminatorias efectivas. Esto empeora cuando aumenta el número de autores.

Inferimos que los escritores tienen diferentes maneras de vincular palabras debido al hecho de que cada escritor hace uso de frases comunes. Además, usan frases en diferentes tasas y, por lo tanto, los textos escritos mantienen una estructura de fondo. Por ejemplo, algunos autores suelen usar la frase "los datos recopilados en el estudio sugieren que [...]" a diferencia de otro autor que utiliza "los datos parecen sugerir que [...]". Por lo tanto, las palabras *the, in, to, that* se pueden incluir en diferentes tópicos puesto que, a diferencia de LSA (Dumais, 2004), LDA puede asignar la misma palabra a diferentes tópicos para manejar mejor la polisemia. Como resultado, el uso de esas palabras en diferentes cantidades dará como resultado diferentes distribuciones de tópicos para cada documento.

Para representar cada problema, todos los documentos en el conjunto de datos se procesan con LDA. Luego, se obtienen vectores (con valores reales, probabilidad de cada tema) que representan documentos conocidos y desconocidos.

Con base en un problema específico, hacemos una resta entre el vector de cada documento conocido y el vector del documento desconocido (recordemos que sólo hay un documento desconocido por problema; sin embargo, existen de uno a cinco documentos conocidos).

Se encontró que la conversión de los valores reales en valores 0, 1 no cambió significativamente los resultados finales, por lo que los convertimos a sus valores binarios utilizando la media aritmética como umbral; 0 representa la ausencia del tópico, y 1, la presencia del mismo. Por lo tanto, la resta entre vectores dio como resultado dos valores

posibles: 0 cuando los tópicos son iguales y 1 cuando los tópicos son diferentes (consulte la figura V.3).

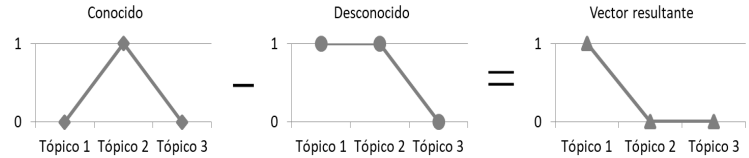


Figura V.3: Ejemplo de resta entre el vector de documento conocido y el vector de documento desconocido

5.3.5. Resultados

En los siguientes experimentos se clasificaron los documentos mediante un algoritmo NB. Además, se especificaron diferentes n-tópicos para LDA. Por lo tanto, se generaron patrones de n características para cada documento.

Se encontró que al variar el número de tópicos, también cambió el rendimiento de la clasificación. Asimismo, no existe un método *a priori* para determinar cuántos tópicos deben ser elegidos para aumentar el rendimiento; por lo tanto, se tuvo que buscar empíricamente un intervalo hasta alcanzar los mejores resultados.

Debido que LDA es un método estocástico, el resultado obtenido para cada experimento puede ser diferente, por lo que se muestra el promedio de 100 experimentos para cada número de tópicos evaluados. Además, se muestra la desviación estándar.

Tenga en cuenta que el promedio de 100 experimentos se calculó para cada medida de forma independiente, es decir, el producto de exactitud

y el área ROC, en las tablas correspondientes, no será proporcional a la medida FS (puntaje final).

La medida FS es el producto de dos valores: c@1 (Reynolds *et al.*, 2000) y el área bajo la curva ROC (AUC) (Fawcett, 2006). La primera es una extensión de la medida de exactitud y la segunda, una medida del rendimiento de clasificación que proporciona resultados más completos que la exactitud.

Suponiendo que, en un caso real, el conjunto de pruebas es desconocido, se realizó, en primera instancia, la validación cruzada de cinco pliegues del conjunto de entrenamiento para encontrar el intervalo óptimo de tópicos, y en segunda fue aplicado el intervalo para evaluar los conjuntos de pruebas.

Tabla V.23: Rendimiento obtenido en el idioma inglés, ajustando el número de tópicos

Tópicos	Valores binarios				Valores originales			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.743	0.733	0.554	0.129	0.495	0.407	0.202	0.019
3	**0.816**	0.853	0.697	0.041	0.672	0.699	0.474	0.090
4	0.800	**0.847**	**0.679**	0.049	0.633	0.637	0.407	0.080
5	0.781	0.835	0.654	0.067	0.631	0.646	0.412	0.085
6	0.759	0.811	0.618	0.075	0.622	0.621	0.389	0.064
7	0.778	0.790	0.586	0.084	0.595	0.598	0.358	0.064
8	0.746	0.800	0.599	0.064	0.591	0.594	0.356	0.083
9	0.735	0.787	0.581	0.075	0.616	0.624	0.388	0.073
10	0.724	0.767	0.559	0.084	0.604	0.603	0.367	0.067

5.3.6. Resultados de la clasificación de PAN 2015

Se muestra en las tablas V.23-V.26, un análisis detallado del rendimiento de la clasificación de cada idioma en la identificación del

Tabla V.24: Rendimiento obtenido en el idioma español, ajustando el número de tópicos

Tópicos	Valores binarios				Valores originales			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.633	0.632	0.402	0.037	0.637	0.672	0.428	0.032
3	0.730	0.765	0.561	0.080	0.673	0.702	0.473	0.046
4	**0.750**	**0.783**	**0.589**	0.066	0.678	0.697	0.474	0.053
5	0.740	0.776	0.576	0.071	0.678	0.698	0.475	0.047
6	0.751	0.777	0.586	0.072	0.661	0.694	0.463	0.061
7	0.754	0.777	0.589	0.075	0.664	0.697	0.465	0.068
8	0.726	0.756	0.551	0.072	0.646	0.676	0.439	0.064
9	0.719	0.747	0.540	0.079	0.639	0.667	0.429	0.070
10	0.715	0.738	0.530	0.075	0.635	0.658	0.420	0.066

Tabla V.25: Rendimiento obtenido en el idioma holandés, ajustando el número de tópicos

Tópicos	Valores binarios				Valores originales			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.722	0.721	0.521	0.030	0.682	0.703	0.479	0.023
3	**0.741**	**0.751**	**0.557**	0.030	0.685	0.710	0.487	0.029
4	0.717	0.736	0.530	0.067	0.671	0.707	0.475	0.035
5	0.675	0.730	0.496	0.067	0.636	0.676	0.432	0.050
6	0.668	0.724	0.485	0.049	0.617	0.661	0.409	0.044
7	0.681	0.730	0.498	0.043	0.619	0.651	0.404	0.046
8	0.690	0.730	0.505	0.041	0.626	0.653	0.410	0.045
9	0.696	0.727	0.508	0.050	0.633	0.660	0.419	0.040
10	0.690	0.715	0.494	0.045	0.629	0.657	0.414	0.038

autor PAN 2015. Como se puede ver, es posible alcanzar el mejor rendimiento configurando diferentes tópicos. Por ejemplo, el método logra el mejor resultado en el caso del inglés cuando establecemos cuatro tópicos (tabla V.23). Para cada tabla, se muestra la exactitud, área ROC, FS, FS-SD (puntaje final-desviación estándar). Además, se muestra una comparación entre el rendimiento obtenido con valores

Tabla V.26: Rendimiento obtenido en el idioma griego, ajustando el número de tópicos

Tópicos	Valores binarios				Valores originales			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.616	0.616	0.384	0.079	0.594	0.611	0.364	0.054
3	0.643	0.674	0.440	0.090	0.564	0.591	0.336	0.060
4	0.668	0.695	0.469	0.087	0.564	0.592	0.336	0.053
5	0.665	0.696	0.466	0.077	0.561	0.599	0.338	0.059
6	**0.671**	**0.709**	**0.479**	0.086	0.567	0.600	0.344	0.075
7	0.661	0.696	0.468	0.114	0.566	0.601	0.344	0.074
8	0.664	0.703	0.471	0.089	0.561	0.595	0.337	0.066
9	0.654	0.690	0.455	0.084	0.585	0.603	0.356	0.071
10	0.669	0.703	0.476	0.100	0.593	0.612	0.366	0.071

originales y valores binarios de LDA; este comparativo sugiere que para el marco establecido en este estudio, los valores binarios obtienen un mejor rendimiento en la clasificación.

Es interesante observar que con los vectores formados para pocos tópicos (1 hasta 10), podemos obtener una exactitud de más del 64 %. De hecho, se puede suponer que los documentos, por una parte, podrían haber sido categorizados por tema; sin embargo, este supuesto es poco probable debido que, como se muestra en la sección 5.3.3.1, el conjunto de datos utilizado está formado por una mezcla de sujetos.

Por otra parte, se llevaron a cabo dos experimentos con el objetivo de determinar si dos documentos escritos por el mismo autor serán similares según su distribución de tópicos. La figura V.4 muestra la suma de todas las diferencias por tópico en el conjunto de datos de prueba para inglés. Como puede verse, el número de diferencias es alto cuando los textos son escritos por diferentes autores. En la figura V.5 se muestran las mismas diferencias para el idioma español.

Se clasificó el conjunto de datos sin preprocesamiento y fueron encontrados los valores mostrados en la tabla V.27. La tabla muestra los valores correspondientes al experimento más cercano al promedio de cada idioma.

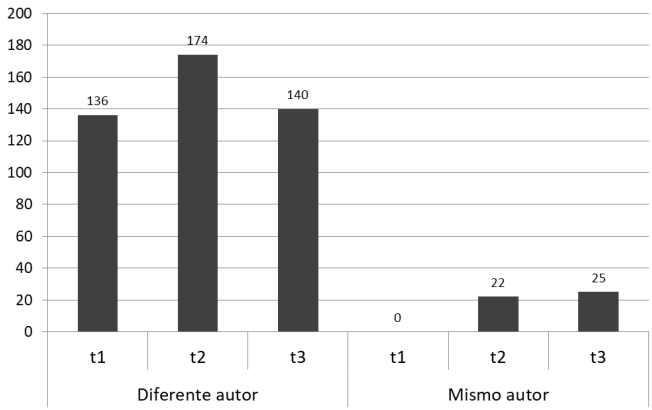


Figura V.4: Diferencias de tópicos entre documentos escritos por el mismo autor o por un autor diferente (subconjunto en inglés)

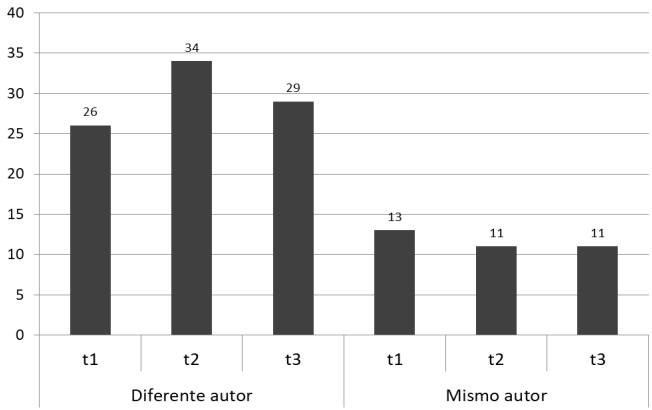


Figura V.5: Diferencias de tópicos entre documentos escritos por el mismo autor o por un autor diferente (subconjunto en español)

De acuerdo con la tabla V.27, se obtuvo el mejor resultado para el subconjunto de inglés con 85.6 % de exactitud, incluso cuando el conjunto de prueba es el más grande (500 problemas) del corpus. El subconjunto de textos en español ocupa el segundo lugar con 76.0 % de exactitud, el subconjunto holandés alcanzó 70.9 % de exactitud y finalmente el subconjunto griego, 64.0 % de exactitud.

Obtuvimos nuestro mejor y peor desempeño con los subconjuntos de inglés y griego respectivamente; por lo tanto, no podemos inferir que la configuración de varios tópicos hizo la diferencia en los resultados, ya que en realidad ambos subconjuntos son del mismo tipo y uno de ellos no se vio afectado. De manera similar, tanto para los subconjuntos españoles como para los holandeses (segundo y tercer lugar, respectivamente), los resultados no llevan a la conclusión de que la mezcla de géneros tiene alguna correlación con ellos.

Fueron comparados los resultados de este estudio con los obtenidos en la tarea de identificación del autor en PAN 2015 (Verhoeven y Daelemans, 2014). Por lo tanto, fueron usadas las mismas medidas de rendimiento que los autores de la tarea PAN 2015.

Se presentan en la tabla V.28 los mejores resultados obtenidos para cada subconjunto de idiomas por los participantes de la tarea PAN 2015. Además, se muestran los peores puntajes FS (puntaje final) de los participantes de PAN 2015. De acuerdo con esos resultados, nuestro método parece funcionar mejor tanto para el idioma inglés como para el holandés. Por una parte, el método propuesto es capaz de superar los resultados de FS con respecto al subconjunto de inglés y tiene un mejor desempeño que Bartoli *et al.* (2015) y el resultado de Bagnall (2015) con respecto al subconjunto holandés.

Por otra parte, para los subconjuntos tanto en español como en griego, el método propuesto no mostró un buen desempeño; sin embargo, nuestros resultados superan los peores puntajes FS de los participantes de PAN 2015 y los resultados de la curva ROC indican que las predicciones son aceptables.

Excluimos que los malos resultados pueden deberse a la longitud del vocabulario, debido a que se obtuvieron 1 000 y 600 tipos en documentos en español y griego, respectivamente; y sólo alrededor de 300 y 200 tipos en documentos en inglés y holandés. Por lo tanto, si el rendimiento dependiera del vocabulario, se obtendrían mejores resultados en los documentos en español y griego.

Inferimos que los resultados dependen de la fuente de los documentos. Es decir, se tomaron artículos en inglés de obras de teatro, que contenían diálogos; en consecuencia, esos documentos pueden mantener cierta estructura latente similar, ya que todos ellos fueron escritos para un propósito específico. A diferencia de los documentos en inglés, los otros tienen diferentes propósitos; por ejemplo, los documentos en español están tomados de periódicos e incluso de *blogs* personales.

Tabla V.27: Resultados de cada clasificación de subconjunto. Exactitud, precisión (P), exahustividad (R) y medida F (F)

Conjunto	Tipo	Exactitud	P	R	F
Inglés	Cross-topic	85.6	0.864	0.856	0.855
Español	Cross-topic/genre	76.0	0.760	0.760	0.760
Holandés	Cross-genre	70.9	0.733	0.709	0.702
Griego	Cross-topic	64.0	0.646	0.640	0.640

Tabla V.28: Comparación de resultados con otros autores. FS=c@1*AUC

Autor	Medida	Subconjuntos			
		Inglés	Español	Holandés	Griego
Bagnall (2015)	c@1	0.757	0.814	0.644	0.851
	AUC	0.811	0.886	0.700	0.882
	FS	0.614	0.721	0.451	0.750
Bartoli *et al.* (2015)	c@1	0.559	0.830	0.689	0.657
	AUC	0.578	0.932	0.751	0.698
	FS	0.323	0.773	0.518	0.458
Moreau *et al.* (2015)	c@1	0.638	0.755	0.770	0.781
	AUC	0.709	0.853	0.825	0.887
	FS	0.453	0.661	0.635	0.693
Este estudio	c@1	0.816	0.750	0.741	0.671
	AUC	0.853	0.783	0.751	0.709
	FS	0.697	0.589	0.557	0.479
Peor resultado PAN'15	FS	0.201	0.095	0.089	0.212

Tabla V.29: Resultados detallados de clasificación en géneros donde se alcanzó la línea de base

Tópicos	Opiniones-holandés		Novelas-inglés		Ensayos-inglés	
	FS	FS-DS	FS	FS-DS	FS	FS-DS
10	0.315	0.041	**0.391**	0.062	0.517	0.042
20	0.312	0.058	0.358	0.051	0.533	0.044
30	**0.325**	0.054	0.333	0.030	0.539	0.040
40	0.301	0.048	0.327	0.054	0.535	0.040
50	0.310	0.066	0.296	0.045	0.540	0.043
60	0.283	0.052	0.292	0.041	**0.551**	0.039
70	0.261	0.050	0.279	0.041	0.540	0.032
80	0.256	0.052	0.282	0.043	0.535	0.024
90	0.281	0.046	0.270	0.038	0.536	0.028
100	0.296	0.051	0.268	0.048	0.546	0.034

Tabla V.30: Resultados detallados de clasificación en géneros donde no se alcanzó la línea de base

Tópicos	Ensayos-holandés		Artículos-griego		Artículos-español	
	FS	FS-DS	FS	FS-DS	FS	FS-DS
10	0.661	0.052	**0.436**	0.050	**0.322**	0.047
20	0.667	0.045	0.406	0.064	0.310	0.056
30	**0.648**	0.053	0.360	0.046	0.280	0.050
40	0.628	0.053	0.352	0.060	0.277	0.054
50	0.595	0.037	0.339	0.042	0.280	0.044
60	0.565	0.059	0.329	0.058	0.261	0.046
70	0.548	0.055	0.304	0.052	0.255	0.045
80	0.544	0.064	0.298	0.039	0.259	0.044
90	0.494	0.057	0.288	0.051	0.275	0.061
100	0.475	0.053	0.267	0.045	0.261	0.047

Tabla V.31: Comparativa de desempeño de este trabajo y líneas de base PAN 2014

Género	c@1	Área ROC	FS de este estudio	FS baseline de PAN 2014
Ensayos-holandés	0.778	0.855	0.667	0.685
Opiniones-holandés	0.560	0.577	0.325	0.322
Ensayos-inglés	0.705	0.780	0.551	0.288
Novelas-inglés	0.605	0.644	0.391	0.202
Artículos-griego	0.631	0.686	0.436	0.452
Artículos-español	0.555	0.576	0.322	0.378

5.3.7. Resultados de la clasificación de PAN 2014

La tarea de PAN 2014 es la misma de PAN 2015 explicada anteriormente; sin embargo, hay una diferencia principal con respecto al conjunto de datos, en PAN 2014 no hay mezcla de dominios. Asimismo, contiene seis subconjuntos, y cada uno consta de textos del mismo género (por ejemplo, ensayos, artículos, novelas).

Tabla V.32: Comparativo de desempeño de este estudio y los participantes de PAN 2014

Género	Mejor puntaje PAN'14	Peor puntaje PAN'14	Este estudio
Ensayos-holandés	0.823	0.307	0.667
Opiniones-holandés	0.525	0.170	0.325
Ensayos-inglés	0.513	0.270	0.551
Novelas-inglés	0.508	0.225	0.391
Artículos-griego	0.720	0.281	0.436
Artículos-español	0.698	0.248	0.322

En esta sección, se indican los resultados de la clasificación PAN 2014. En las tablas V.29 y V.30 se muestran los resultados detallados para los subconjuntos en los que la línea de base de PAN 2014 fue y no fue alcanzada, respectivamente. En este caso, se superó la línea de base, establecida por el desafío PAN 2014, para las revisiones holandesas, las novelas en inglés y los ensayos en inglés (tres de los seis subconjuntos, consulte la tabla V.31).

En la tabla V.32 se muestran los mejores resultados obtenidos por los participantes del PAN 2014. Nuestro método sólo supera en clasificación de ensayos en inglés con respecto a las mejores puntuaciones; sin embargo, no está por debajo de los peores.

Conclusión

En este libro se mostró la aplicación de un espacio semántico continuo para analizar principalmente la tarea de identificación de texto engañoso; de igual manera, se analizaron tres casos de estudio, en donde los mejores resultados se consiguieron clasificando documentos de un mismo dominio o tema; también, se obtuvieron los mejores resultados usando LDA. En los casos restantes (dominios mezclados y cruzados), los métodos de generación de características con los mejores resultados fueron variados.

Se ha mostrado que las características generadas mediante un modelo de espacio semántico continuo han complementado en mayor o menor grado aquellas generadas mediante otros enfoques; aunque se ha probado que cuando se escribe un texto para engañar a otra persona, en la mayoría de los casos, tendemos a cambiar la forma en que seleccionamos las palabras. Se mostró, además, que las palabras representantes de cada clase (texto veraz y engañoso) no serán estrictamente definitivas de una clase en particular en todos los casos de estudio.

Esto podría sugerir que es mejor optar por un método que capture las características representativas de los documentos, antes que un método que busque palabras universales para detectar engaño, o en cambio, usar los algoritmos de aprendizaje automático, aplicarlos a un conjunto de datos enorme y encontrar un conjunto de palabras o fenómenos

lingüísticos que discriminen fuertemente una u otra clase. Sin embargo, etiquetar el texto engañoso y veraz es una tarea retórica, por lo que los estudios, por el momento, se limitan a estudios en un dominio específico. Por lo general, la mayoría de los trabajos en detección de engaño usan y combinan diferentes métodos para generar características y así mejorar la detección de engaño; sin embargo, no tienen en cuenta el hecho de que las características pueden cambiar según la naturaleza del texto.

Referencias

Afroz, Sadia, Michael Brennan y Rachel Greenstadt (2012).“Detecting hoaxes, frauds, and deception in writing style online” en *2012 IEEE Symposium on Security and Privacy*, IEEE, pp. 461-475.

Almela, Ángela, Rafael Valencia García y Pascual Cantos (2012). “Seeing through deception: A computational approach to deceit detection in written communication” en *Proceedings of the workshop on computational approaches to deception detection*, Association for Computational Linguistics, pp.15-22.

Bagnall, Douglas (2015). “Author identification using multi-headed recurrent neural networks”. Disponible en https://arxiv.org/ftp/arxiv/papers/1506/1506.04891.pdf, consultado en abril de 2020.

Bartoli, Alberto *et al.* (2015). “An author verification approach based on differential features” en *Conference and Labs of the Evaluation forum*, vol. 1391, CEUR.

Bergsma, Shane, Matt Post y David Yarowsky (2012). “Stylometric analysis of scientific articles” en *Proceedings of the 2012 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 327-337. Disponible en https://www.aclweb.org/anthology/N12-1033.pdf, consultado en abril de 2020.

Blei, David M., Andrew Y. Ng y Michael I. Jordan (2003), "Latent dirichlet allocation" en *Journal of Machine Learning Research*, vol. 3, pp. 993-1022.

Bradley, Joseph K., Patrick Gage Kelley y Aaron Roth (2008). "Author identification from citations". Disponible en http:www.cs.cmu.edu epxingClass1070106fproject-reportsbradley_kelley_roth.pdf, consultado en abril de 2020.

Castro, Antonio y Brian Lindauer (2012). "Author identification on twitter". Disponible en http://cs229.stanford.edu/proj2012/CastroLindauer-AuthorIdentificationOnTwitter.pdf, consultado en mayo de 2020.

Chomsky, Noam (1956). "Three models for the description of language" en *IRE Transactions on information theory*, vol. 2, pp. 113-124.

DePaulo, Bella M. *et al.* (2003). "Cues to deception" en *Psychological bulletin*, vol. 129, p. 74.

Douglas, Dan y Kathleen M. Broussard (2000). "Longman grammar of spoken and written english" en *Tesol Quarterly*, vol. 34, pp. 787-788.

Dumais, Susan T. (2004), "Latent semantic analysis" en *Annual Review of Information Science and Technology*, vol. 38, pp. 188-230.

Ekman, Paul (1989). “Why lies fail and what behaviors betray a lie” en *Credibility assessment*, Springer, pp. 71-81.

Ekman, Paul (2009). *Telling lies: Clues to deceit in the marketplace, politics, and marriage* (revised edition), WW Norton & Company, New York.

Fawcett, Tom (2006). “An introduction to roc analysis” en *Pattern recognition letters*, vol. 27, pp. 861-874.

Fellbaum, Christiane (1998). “Towards a representation of idioms in WordNet” en *Proceedings of the workshop on the Use of WordNet in Natural Language Processing Systems* (Coling-ACL 1998), Montreal, Canada.

Feng, Song, Ritwik Banerjee y Yejin Choi (2012). “Syntactic stylometry for deception detection” en *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Association for Computational Linguistics*, vol. 2, pp. 171-175.

Fornaciari, Tommaso y Massimo Poesio (2012). “On the use of homogenous sets of subjects in deceptive language analysis” en *Proceedings of the Workshop on Computational Approaches to Deception Detection*, Association for Computational Linguistics, pp. 39-47.

Fornaciari, Tommaso y Massimo Poesio (2014). “Identifying fake amazon reviews as learning from crowds” en *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 279-287.

Fusilier, Donato Hernández *et al.* (2015). "Detection of opinion spam with character n-grams" en *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, pp. 285-294.

Gokhman, Stephanie *et al.* (2012). "In search of a gold standard in studies of deception" en *Proceedings of the Workshop on Computational Approaches to Deception Detection, Association for Computational Linguistics*, pp. 23-30.

Green, Rachel M. y John W. Sheppard (2013). "Comparing frequency-and style-based features for twitter author identification" en *The Twenty-Sixth International FLAIRS Conference*. Disponible en https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5917/6043, consultado en mayo de 2020.

Guyon, Isabelle y André Elisseeff (2003). "An introduction to variable and feature selection" en *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182.

Hall, Mark Andrew (1999). "Correlation-based feature selection for machine learning". Disponible en https://www.cs.waikato.ac.nz/ mhall/thesis.pdf, consultado en febrero de 2020.

Hall, Mark *et al.* (2009), "The weka data mining software: an update" en *ACM SIGKDD explorations newsletter*, vol. 11, pp. 10-18.

Hauch, Valerie *et al.* (2012). "Linguistic cues to deception assessed by computer programs: A meta-analysis" en *Proceedings of the workshop on computational approaches to deception detection, Association for Computational Linguistics*, pp. 1-4.

Hernández Castañeda, Ángel e Hiram Calvo (2017). "Deceptive text detection using continuous semantic space models" en *Intelligent Data Analysis*, vol. 21, pp. 679-695.

Hu, Minqing y Bing Liu (2004). "Mining and summarizing customer reviews" en *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, pp. 168-177.

Indurkhya, Nitin y Fred J. Damerau (2010). *Handbook of natural language processing*, vol. 2. CRC Press.

Kamp, Hans (1981). "A theory of truth and semantic representation" en *Formal semantics -the essential readings*, pp. 189-222.

Keila, P. S. y David B. Skillicorn (2005). "Detecting unusual email communication" en *Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research* , IBM Press, pp. 117-125

Layton, Robert, Paul Watters y Richard Dazeley (2013). "Local ngrams for author identification" en *Notebook for PAN at CLEF 2013*. Disponible en https:pan.webisḋedownloadspublicationspaperslayton_2013ṗdf, consultado en febrero de 2020.

Madigan, David *et al.* (2005). "Author identification on the large scale" en *Proceedings of the Meeting of the Classification Society of North America*, vol. 13.

Marneffe de, Marie-Catherine *et al.* (2006). "Generating typed dependency parses from phrase structure parses" en *Lrec*, vol. 6, pp. 449-454.

Masip, Jaume *et al.* (2012). "Deception detection from written accounts" en *Scandinavian Journal of Psychology*, vol. 53, pp. 103-111.

Mihalcea, Rada y Carlo Strapparava (2009). "The lie detector: Explorations in the automatic recognition of deceptive language" en *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, Association for Computational Linguistics*, pp. 309-312.

Mohtasseb, Haytham, UK Lincoln y Amr Ahmed (2009). "More blogging features for author identification". Disponible en file:///C:/Users/Usuario/Downloads/MoreBloggingFeaturesforAuthorIdentification.pdf, consultado em febrero de 2020.

Moreau, Erwan *et al.* (2015). "Author verification: basic stacked generalization applied to predictions from a set of heterogeneous learners-notebook for pan at clef 2015" en *CLEF 2015-Conference and Labs of the Evaluation forum*, CEUR.

Narayanan, Arvind *et al.* (2012). "On the feasibility of internet-scale author identification" en *2012 IEEE Symposium on Security and Privacy*, IEEE, pp. 300-314.

Newman, Matthew L. *et al.* (2003). "Lying words: Predicting deception from linguistic styles" en *Personality and social psychology bulletin*, vol. 29, pp. 665-675.

Nirkhi, Smita y Rajiv V. Dharaskar (2013). "Comparative study of authorship identification techniques for cyber forensics analysis" en *International Journal of Advanced Computer Science and Aplicattions*, vol. 14, núm. 5, pp. 32-35

Ott, Myle, Claire Cardie y Jeffrey T Hancock (2013). "Negative deceptive opinion spam" en *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 497-501.

Ott, Myle *et al.* (2011). "Finding deceptive opinion spam by any stretch of the imagination" en *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 309-319.

Pacheco, María Leonor, Kelwin Fernandes y Aldo Porco (2015). "Random forest with increased generalization: A universal background approach for authorship verification" en *CLEF (Working Notes).*

Pang, Bo, Lillian Lee y Shivakumar Vaithyanathan (2002). "Thumbs up?: sentiment classification using machine learning techniques" en *Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Association for Computational Linguistics*, vol. 10, pp. 79-86.

Pateriya, Pushpendra Kumar *et al.* (2012). "A study on author identification through stylometry" en *International Journal of Computer Science & Communication Networks*, vol. 2, p. 653.

Pavelec, Daniel, Edson Justino y Luiz S. Oliveira (2007). "Author identification using stylometric features" en *Inteligencia Artificial*. Revista Iberoamericana de Inteligencia Artificial, vol. 11.

Pennebaker, James W. *et al.* (2015), "The development and psychometric properties of liwc2015". *Reporte técnico*. Disponible en file:C:UsersUsuarioDownloadsLIWC2015_LanguageManual.pdf, consultado en febrero de 2020.

Pérez Rosas, Verónica y Rada Mihalcea (2014a). "Cross-cultural deception detection" en *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (vol. 2: Short Papers), vol. 2, pp. 440-445.

Pérez Rosas, Verónica y Rada Mihalcea (2014b), "Gender differences in deceivers writing style" en *Mexican International Conference on Artificial Intelligence*, Springer, pp. 163-174.

Pérez Rosas, Verónica *et al.* (2015). "Verbal and nonverbal clues for real-life deception detection" en *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2336-2346.

Pimas, Oliver, Mark Kröll y Roman Kern (2015). "Know-center at pan 2015 author identification" en *Working Notes Papers of the CLEF*.

Pustejovsky, James (1991). "The generative lexicon" en *Computational Linguistics*, vol. 17, pp. 409-441.

Raykar, Vikas C. *et al.* (2010). "Learning from crowds" en *Journal of Machine Learning Research*, vol. 11, pp. 1297-1322.

Rayson, Paul, Andrew Wilson y Geoffrey Leech (2002). “Grammatical word class variation within the british national corpus sampler” en *New Frontiers of Corpus Research*, Brill Rodopi, pp. 295-306.

Reynolds, Douglas A., Thomas F Quatieri y Robert B. Dunn (2000). “Speaker verification using adapted gaussian mixture models” en *Digital signal processing*, vol. 10, pp. 19-41.

Savoy, Jacques (2013). “Authorship attribution based on a probabilistic topic model” en *Information Processing & Management*, vol. 49, pp. 341-354.

Schelleman Offermans, Karen y Harald Merckelbach (2010). “Fantasy proneness as a confounder of verbal lie detection tools” en *Journal of Investigative Psychology and Offender Profiling*, vol. 7, pp. 247-260.

Seroussi, Yanir, Ingrid Zukerman y Fabian Bohnert (2014). “Authorship attribution with topic models” en *Computational Linguistics*, vol. 40, pp. 269-310.

Sidorov, Grigori *et al.* (2014). “Syntactic n-grams as machine learning features for natural language processing” en *Expert Systems with Applications*, vol. 41, pp. 853-860.

Stamatatos, Efstathios (2009). “A survey of modern authorship attribution methods” en *Journal of the American Society for information Science and Technology*, vol. 60, pp. 538-556.

Stamatatos, Efstathios *et al.* (2015). “Overview of the pan/clef 2015 evaluation lab” en *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, pp. 518-538,

Stamatatos, Efstathios *et al.* (2014). "Overview of the author identification task at pan 2014" en *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers*, Sheffield, UK, 2014, pp. 1-21.

Toma, Catalina L. y Jeffrey T. Hancock (2012). "What lies beneath: The linguistic traces of deception in online dating profiles" en *Journal of Communication*, vol. 62, pp. 78-97.

Twitchell, Douglas P, Jay F. Nunamaker y Judee K. Burgoon (2004). "Using speech act profiling for deception detection" en *International Conference on Intelligence and Security Informatics*, Springer, pp. 403-410.

Verhoeven, Ben y Walter Daelemans (2014). "Clips stylometry investigation (CSI) corpus: a dutch corpus for the detection of age, gender, personality, sentiment and deception in text" en *LREC 2014-ninth International Conference on Language Resources and Evaluation*, pp. 3081-3085.

Vossen, Piek (1998). "Introduction to eurowordnet" en *EuroWordNet: A multilingual database with lexical semantic networks*, Springer, Dordrecht, pp. 1-17.

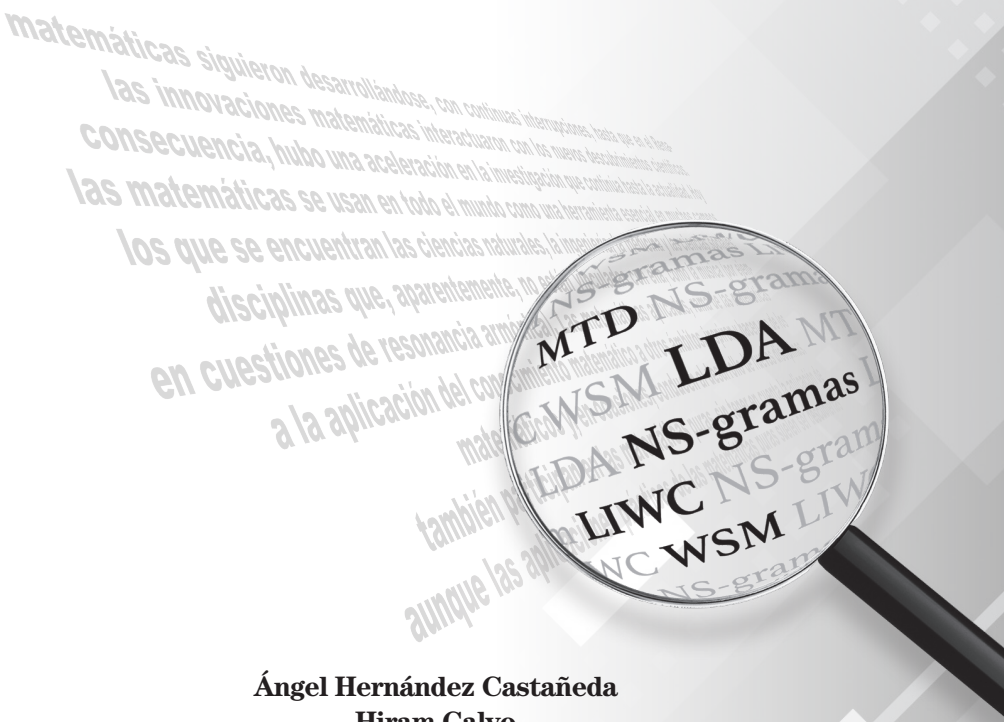
Vossen, Piek (2001). "Extending, trimming and fusing WordNet for technical documents" en *Proceedings on NAACL2001 workshop on WordNet and other lexical resources applications, extensions and customizations*, Pittsburgh, USA June 2001. The Association for Computational Linguistics.

Wierzbicka, Anna (1972). *Semantic primitives, Frankfurt, Athenäum-Verl*, serie: Linguistische Forschungen.

Williams, Shanna Mary *et al.* (2014). "Is the truth in your words? distinguishing children's deceptive and truthful statements" en *Journal of Criminology*, pp.1-9.

Xu, Qiongkai y Hai Zhao (2012). "Using deep linguistic features for finding deceptive opinion spam" en *Proceedings of Coling 2012: Posters*, pp. 1341-1350.

Automatic detection of deceptive text by means of a continuous semantic space model



Ángel Hernández Castañeda
Hiram Calvo
René Arnulfo García Hernández

Luis Cejudo Espinosa
Translator

Toluca, 2020

Universidad Autónoma del Estado de México

Alfredo Barrera Baca, Ed.D.
Rector

Carlos Eduardo Barrera Díaz, D.Env.
Secretary of Research and Advanced Studies

Martín Carlos Vera Estrada, Eng.D.
Coordinator of Tianguistenco Professional Academic Unit

Susana García Hernández, MBA
Director of Dissemination and Promotion
of Research and Advanced Studies

Automatic detection of deceptive text by means of a continuous semantic space model

Automatic detection of deceptive text
by means of a continuous semantic space model

Ángel Hernández Castañeda
Hiram Calvo
René Arnulfo García Hernández

First edition: november, 2020
ISBN 978-607-633-222-1 (electronic)
ISBN 978-607-633-221-4 (print)

D. R. © Universidad Autónoma del Estado de México
Instituto Literario núm. 100 Ote.
C.P. 50000, Toluca, Estado de México
www.uaemex.mx

Front cover: Juan Manuel García Guerrero
Translator: Luis Cejudo Espinosa

The present book was reviewed and approved by two double blind peers, attached to the National System of Researchers, alien to Universidad Autónoma del Estado de México. The arbitration was in charge of the Secretary of Research and Advanced Studies, as verified in file 234/2019.

The present edition and its characteristics are property of Universidad Autónoma del Estado de México.

The content in this publication is the authors' responsibility.

Funds from Program for the Strengthening of Education Excellence (Profexce), are public in nature; its use for political purposes or personal gain is forbidden.

Index

Introduction **162**

1. Detection of deceptive text **164**

1.1. Why is it important to detect deception in texts? 168
1.2. Identifying deceptive text is a great challenge 169

2. Basic concepts **172**

2.1. Text processing 172
2.2. Tokenization 174
2.3. Parsing 176
2.3.1. Context-Free Grammars 177
2.3.2. Parse trees 178
2.3.3. Flexibility 179
2.3.4. Disambiguation 181
2.4. Semantic analysis 182
2.4.1. Theories and approaches to semantic representation 184
2.4.2. Logical approaches 184
2.4.3. Discourse representation theory 185
2.4.3.1. Pustejovsky's generative lexicon 186
2.4.3.2. Natural semantic metalanguage 187
2.4.3.3. Object-oriented semantics 188
2.4.3.4. Relational facts and ontologies 189

3. Methods to identify deception **190**

3.1. Background 190
3.2. Detection of deception in controversial topics 193
3.3. Detection of deception in opinions about hotels 197
3.4. Detection of deception in statements 200
3.5. Detection of deception in combinations of domains 201
3.6. Detection of deception in opinions on books 205

4. A new approach to deception detection **210**

4.1. Sources of generation of characteristics 210
4.1.1. Term-document matrix (TDM) 211
4.1.2. Syntactic n-grams 212
4.1.3. Linguistic Inquiry and Word Count 214

4.1.4. Latent Dirichlet Allocation (LDA) – continuous semantic space model 216
4.2. Datasets 218
4.2.1. DeRev dataset 219
4.2.2. OpSpam dataset 220
4.2.3. Dataset of controversial topics 222
4.2.4. Dataset analysis 223
4.3. Construction of vectors of characteristics 224
4.4. Identification of the optimal number of topics 227

5. Assessment of the approach proposed 230

5.1. Detection of deception 230
5.1.1. Validation measures 231
5.1.1.1. F-measure 232
5.1.1.2. Area under the ROC curve 233
5.1.2. Classification of a specific domain 234
5.1.3. Classification of mixed domains 237
5.1.4. Crossed domain classification 239
5.1.5. Prevailing words and topics 240
5.1.6. Comparison of results and statistical significance 245
5.2. Impact of polarity on deceptive texts 247
5.2.1. Polarity classification to improve deception detection 254
5.3. Solving a different task 257
5.3.1. Authorship identification 257
5.3.2. Approaches to verify authorship 259
5.3.3. Datasets 261
5.3.3.1. PAN 2015 dataset 262
5.3.3.2. PAN 2014 dataset 263
5.3.4. Method proposed 263
5.3.4.1. Method to verify authorship 264
5.3.5. Results 267
5.3.6. Results of PAN 2015 classification 269
5.3.7. Results of PAN 2014 273

Conclusion 276
References 278
About the authors 290

Index of figures

II.1. Example of a parse tree 180

IV.1. Example of vector construction using a model of space of words 214

IV.2. Process to generate syntactic bigrams from a sentence 215

IV.3. Example of some groups of words in LIWC with their tag. A group with only seven words per tag is presented 216

IV.4. Example of topic generation, from texts on *capital punishment*, by means of LDA 217

IV.5. Example of a document processed by LDA. The distribution of topics is shown 218

V.1. Precision and recall 233

V.2. Learning curve for the various corpora 237

V.3. Example of subtraction of the known document vector and the unknown document vector 267

V.4. Topic differences between documents written by the same or a different author (English subset) 271

V.5. Topic differences between documents written by the same or a different author (Spanish subset) 271

Index of tables

III.1. Dominant LIWC classes of deceptive and truthful text 196

III.2. Accuracy by topic using a 10-fold cross validation 196

III.3. Accuracy, precision (P), recall (R), y F-measure (F) of man-made deceptive text identification 199

III.4. Results of the application of various generation sources of characteristics and application of two classifiers 199

III.5. Results of the classification of corpus DeCours, showing: precision (P), recall (R), F-measure (F) and the mean of F-measures as well 202

III.6. Experiments on each country's corpora, separately 203

III.7. Culture classification crossed with LIWC and unigrams as sources of characteristics 204

III.8. Prevailing LIWC classes in deceptive and truthful texts 206

III.9. Classification of DeRev that shows the performance of the tagging methods 208

IV.1. Comparison between syntactic and conventional bigrams in authorship identification 213

IV.2. Comparison between syntactic and conventional trigrams in authorship detection 213

IV.3. Amount of types and *tokens* in the corpora. DTA= Document token average 223

IV.4. Average of types shared by the corpora 223

IV.5. Comparison of accuracy (%) regarding binary values and attribute selection (AtSe) 227

IV.6. F-measure using different values for number of topics 227

V.1. OpSpam classification by means of various sources of generations of characteristics 238

V.2. Number of relevant characteristics obtained from the combination of LDA and TDM 238

V.3. DeRev classification by means of various sources of generation of characteristics 239

V.4. Classification of *abortion* using various sources of generation of characteristics 240

V.5. *Best friend* classification by using various sources of generation of characteristics 241

V.6. *Capital punishment* classification using various sources of generation of characteristics 242

V.7. Accuracy, precision (P), recall (R) and F-measure (F) obtained with the corpora mixed by means of SVM 242

V.8. Accuracy, precision (P), recall (R) and F-measure (F) obtained with the corpora mixed by means of NB 243

V.9. F-measure from the cross-domain classification. Results for NB and SVM are displayed 243

V.10. The 10 most relevant topics and words, representing deceptive and truthful texts in OpSpam 245

V.11. The 10 most relevant topics and words, representing deceptive and truthful texts in *best friends* 246

V.12. The 10 most relevant topics and words, representing deceptive and truthful texts in *abortion* 247

V.13. The 10 most relevant topic and words, representing deceptive and truthful texts in *capital punishment* 248

V.14. The 10 most relevant topics and words, representing deceptive and truthful text in DeRev 249

V.15. Comparison of our results with other studies on the same corpora 250

V.16. Statistical significance 251

V.17. Results of the classification excluding polarity words 253

V.18. Classification of OpSpam, tagged by the authors (OAP) and that of the present study (PTW), based on the text's polarity 253

V.19. Classification of the datasets adding the polarity feature 254

V.20. Comparison of F-measure before (F-Ant) and after (Avg. F) polarity classification. Precision (P), Recall (R), F-measure (F) and the number of opinions by class (# docs.) are displayed as well 254

V.21. Overview of the dataset for PAN 2015 author identification task 261

V.22. Overview of the dataset for PAN 2014 author identification task 262

V.23. Performance in English, adjusting the number of topics 268

V.24. Performance in Spanish, adjusting the number of topics 268

V.25. Performance in Dutch, adjusting the number of topics 269

V.26. Performance in Greek, adjusting the number of topics 269

V.27. Results of each subset classification. Accuracy, precision (P), recall (R) and F-measure (F) 273

V.28. Comparison of results with other authors'. FS=c@1*AUC 273
V.29. Details of the results from genre classification in which baseline was reached 274
V.30. Details of the results from genre classification in which baseline was not reached 274
V.31. Comparison of performance, this work and PAN 2014 baseline 275
V.32. Comparison of performance, this study's and PAN 2014 participants 275

Introduction

Artificial intelligence (AI) is a field of study that intends to replicate the cognitive processes human beings are able to perform in order to incorporate them into computers. A particular research line of such field is Natural Language Processing (NLP), which has as an objective to facilitate communication between computers and human language.

Arising from a doctoral thesis, this book presents a new method to identify deception in texts by means of NLP techniques and machine learning logarithms. It is aimed at students and researchers interested in AI and NLP, specifically to produce models to detect deceptive text.

The overall processes entailed in the approach proposed in this text are as follows: by means of NLP tools, the texts are converted into a vector space, this is to say, vectors of characteristics are generated. Later on, several selection methods to discriminate characteristics of little relevance were applied. Eventually, the resulting vectors were sent to a supervised classification algorithm that identified if they were deceptive or truthful.

It is worth mentioning that this research's goals were to experiment with models based on topics, particularly LDA (Latent Dirichlet Allocation) with a view to detecting deception in the texts under study; in like manner, to combine characteristics generated from a number of

approaches so as to improve efficiency in deception detection. Assessing, at once, the performance of selected approaches (on its own or combined) in a number of case studies and that of methods and combinations of them in the classification of specific, mixed and crossed domains, the impact of polarity in classification of deceptive texts and the application of the proposed approach to other NLP areas.

Owing to the above, the present book is divided into five chapters: in the first one, there is a brief introduction to deceptive text and the experimental approaches of this research are presented. In the second, the theoretical foundation to understand the tools to generate the characteristics applied is presented. The third chapter presents a number of works on various approaches to identify deceptive text by means of machine learning. In the fourth, the approach proposed in this text together with analyzed and classified datasets are studied. Chapter five, last one, describes the results obtained in various cases analyzed.

Chapter 1

Detection of deceptive text

The study of deception has perfected over time, from face analysis, for example, by means of eye movements and face expressions, to the monitoring of human beings' physiologic responses through a polygraph. This last approach has accomplished optimal results (with accuracy over 95%) in deception detection. The operating principles of these deception detectors are mainly based on the following hypothesis: lying produces certain emotional states and for each emotional state there is a physiologic reaction identifiable by the deception detector revising variables such as blood pressure, heart and breathing rates.

In like manner, NLP researchers have tried to identify deception in a text because when people try to deceive in writing, they use certain words that they would not use in a truthful text; that is to say, deception propitiates an emotional state that changes the way we write. For example, if we are performing a special task and we change from a normal state to a stressful discussion, it is possible that we do not remember or omit certain details of the ongoing task because of the stress we are under.

Writing a deceptive text for an individual or a group of people may produce, consciously or unconsciously, a change in behavior; that is to say, in the usual way we carry out a task. This change takes place because the brain is always ready to tell the truth; however in order to lie, it must increase the performance of cognitive processes.

In this regard, Ekman (2009: 26) mentions: "A liar can choose not to lie. Deceive a victim is a deliberate action; a liar intends to misinform the victim. A liar may be justified or not, from their standpoint or the community's. A liar may be a good or bad individual who has chosen to deceive or be truthful, and knows the difference between them".

A philosophical debate on what the concept of *lying* means may be encouraged (since there is no universal definition); for this study's goals though, it considered that deception occurs when a liar intends to transmit fully or partly false information to a victim, i.e., make the victim believe a fact that was not or took place differently. Likewise, the improvement of deception detection in opinions on a number of domains was also sought.

Nowadays, users carry out a number of online activities such as buying and selling goods, disseminate ideas over blogs and interchange general information, which as such is not always reliable; some people use the Internet to transmit information in order to manipulate and deceive other users. This way, when a user wants to make a purchase online, the usual way to find out whether a product is worth is to read the opinions on the seller's website. Those opinions have proven to be decisive on the final decision to buy or not the item. Owing to this, some salespeople hire individuals, sometimes experts, to write positive reviews with the purpose of increasing a product's sales, even if such people do

not have an actual opinion on the qualities and attributes of the product; on other cases, deceptive opinions try to discredit the products offered by competitors.

In addition to deceptive texts written to manipulate the users' purchase decisions, there are also deceptive texts aimed at changing people's opinion or viewpoint on a certain topic, for instance a political candidate or a public debate. All of this makes it very important to advance in the detection of deceptive texts; the task can be basically defined as the identification of written opinions in which the author tries to disseminate information they do not believe (Keila and Skillicorn, 2005).

Studies on deceptive texts have empirically demonstrated that truthful communication is qualitatively different from misleading (Ekman, 1989; Twitchell *et al.*, 2004). Owing to this, various projects have been launched in order to identify deceptive texts as accurately as possible. To do so, various datasets that comprise texts labeled, either truthful or deceptive, have been produced to allow measuring the efficiency of approaches proposed to detect deception.

In the scope of machine learning, some of the texts in the datasets are utilized as a training set for a classifier and the rest as a testing set. Therefore, a direct comparison between various classifiers and methods to select characteristics is possible by means of applying them to the same dataset.

In order to try the efficiency in detecting deceptive text, three datasets on a number of topics were processed, specifically: OpSpam, opinions on hotels; DeRev, opinions on books purchased on Amazon;

and, a dataset of controversial topics, composed of opinions on *abortion*, *capital punishment* and *best friend*.

Various case studies were analyzed. In the first place, each dataset was assessed separately (section 5.1.2) in a design in which the texts refer to one same topic or domain; secondly, experiments mixing all the datasets were run (section 4.2) to verify if the characteristics from a mix of domains (section 5.1.3) improved the results or if they worsened them. Finally, it was enquired if a set of characteristics may be sufficiently general to be used in the classification of a dataset that deals with a topic other than that of the dataset utilized to produce it, which allows creating general-purpose deceptive text detectors independent from domain (section 5.1.4).

To make use of a supervised classifier, it is necessary to map texts into numerical vectors; to do so, characteristics were produced by means of various methods such as Latent Dirichlet Allocation (LDA), Linguistic Inquiry and Word Count (LIWC), and a Word Space Model (WSM), and combinations of characteristics generated by the utilized methods.

On the basis of the available datasets, it was researched what method is the best for a single domain, in which both the training and testing sets share a common topic; for a mix-domain environment, in which the training and testing sets are a mix of topics; and, for a crossed-domain environment where the training and testing sets comprise various topics (with no interception between testing and training).

Running these experiments allowed exploring the possibility to use an existing dataset to detect deceptive texts for which there is no available dataset; that is to say, the possibility to develop a deceptive text detector independent from domain.

1.1. Why is it important to detect deception in texts?

Presently, selling products and services is moving toward the web. Unlike a sale in which the product is physically present to look at, on an online sale there is only one set of images to look through the potential purchase.

Sellers considered that images were not enough for the buyers to be sure that the product will meet their expectations; thereby, they created a section in which buyers can add a comment after receiving and using the product. These were useful for prospective clients to ponder their purchase intention; if opinions are largely positive the buyer will know the product is good; otherwise, if there are too many negative opinions, the buyer will have the information to decide whether to buy the product or not.

This was a good idea until salespeople turned it into a way to take an advantage and increase their business. As a result, various websites that offer their products look for people to write deceptive reviews for them; such opinions will express the products are very good, when in reality the person who wrote it had the intention to help sellers increase their incomes.

Misleading opinions are not necessarily a positive review for the product. It may be the case that competitors want to disparage a certain seller, so comments that will not favor the product in question are going to appear; such sort of opinions will be negative and deceptive at once.

In an article[1] published in *The Economist* in 2015, it was concluded that more than 1 000 users in Amazon[2] engage in writing deceptive opinions. In a particular case, according to this article, a restaurant with one more star in Yelp (a popular review website) increases its incomes from 5 to 9 percent. Not only Amazon, many other websites have these sort of opinions; the motivation is clear: increase sales.

The dissemination of this sort of deceptive opinions is mainly due to human beings' problems to detect deception. This is evinced in the research by Pérez Rosas *et al.* (2015), in which the performance of three human judges in detecting deception in text, audio and video was evaluated. Likewise, it is demonstrated that performance ranges are 54.2-65.3%, 58.5-70.3%, and 63.0-71.0% for text, audio and video, respectively. It is noticed that the most difficult case to differentiate deception from truth is text; this may be because we do not see the individual who is lying, thereby certain information cannot be obtained; for example, if the subject is nervous, speaks loudly, sweats or gives a signal to improve performance in deception detection.

Due to the foregoing, research on new approaches that allow detecting deception in texts more efficiently is necessary. This way, mechanisms that help buyers make safer and more reliable purchases online will be provided.

1.2. Identifying deceptive text is a great challenge

Detecting deception in a text is even a greater challenge than detecting it in a person, since there are only the words in the document or

[1] http://www.economist.com/news/business/21676835-evolving-fight-against-sham-reviews-five-star-fakes

[2] www.amazon.com

collection of documents; that is to say, we lack the individual to monitor their physiological reactions, face expressions and behavior.

A number of studies have tried to look for words that identify deceptive text; however, the following questions may be asked: Are these words the same for different sorts of deception? Can the words that identify deceptive text change if the topic the texts refer to changes as well? What is the method to produce the characteristics generated by a model that best represents a deceptive text? Is the combination of various methods able to produce a better result in detecting deceptive text?

It is difficult to decide whether people have a predefined behavior as they carry a certain action such as lying. Albeit, on the basis of several previously-undertaken works (section 3), it has been identified that liars follow a use pattern for some words; for example, when lying they tend to be unspecific or use more o less certain words.

These research works empirically verify that by means of pattern recognition algorithms it is possible to distinguish a deceptive text from a truthful one, even though there are other restrictions that still have to be analyzed, for example, lack of datasets (or commonly small sets), which does not allow the proposed approaches to be tried in a more general environment.

Chapter 2

Basic concepts

2.1. Text processing

While natural language is inherently ambiguous, written language can be even more ambiguous (Indurkhya *et al.*, 2010).

Text processing is the task of turning an amount of text files into a well-defined sequence of significant linguistic units. Text processing is an essential part of any textscnlp system as the characters, words and sentences identified at this stage are the fundamental units that go on to the next processing stage. These phases may be used for analyzing or tagging, for instance, morphologic analysis and tagging of grammar categories by means of applications such as information retrieval and automatic translation systems.

Text processing may be divided into two phases. Firstly, the process of turning a set of digital files into a well-defined set of texts. This entails, for its part, defining the algorithms to analyze the documents' specific language; identifying the language defines then the natural language in which the documents are produced. This step is closely related to character codification (this on its own does not define it, though).

The second phase is text segmentation; this is the process of turning a well-defined-text corpus into components such as words and sentences. Segmentation consists in severing or trimming the character sequences in the text by means of locating the words' limits, i.e., the points where a word starts and finishes. For the ends of computational linguistics, the words identified in text segmentation are called tokens; therefore, the segmentation process is also known as tokenization.

On one side, text normalization is a related process, which includes all the written variations of a token in a normalized canonical form; for instance: *Mister*, *Mr.*, *mister*, *MISTER*. On the other, the process of sentence segmentation can also be included; this process identifies the beginning and end of the sentences.

The sort of writing used by a language is the most important factor to define the best preprocessing approach. Writing systems may be logographic, in which a vast number of individual symbols represent words; syllabic, in which individual symbols account for syllables; alphabetic, in which individual symbols represent sounds. In practice, there are no modern writing systems that use only one sort of symbols; consequently, no natural-language writing system may be deemed purely logographic, syllabic or alphabetic. English language, for example, is based on the Roman alphabet and uses logographic symbols, including Arabic numerals, currency symbols, among others; although the last of the aforementioned systems is predominantly alphabetic, many of the other writing systems are largely composed of a single sort of symbols.

Tokenization is inescapably related to the subjacent codification of characters in the text to be processed, and character identification is a first essential step. While the heading of a digital document may

contain the corresponding information with its character codification, this information is not always included or even available; in this case, codification must be automatically defined.

In addition to the variety of sorts of symbols (logographic, syllabic or alphabetic) used in writing systems, there exists a set of orthographic conventions in written languages to indicate the limits between linguistic units such as syllables, words or sentences. English resorts to spaces between words and punctuation marks as sentence limits; however, even this is not enough to entirely segment the text resolving the ambiguities.

Many of the existing algorithms to segment natural languages are for specific languages and depend on corpora developed to manage predictable ambiguities in a well-formed text. Depending on the text's origin and purpose, the use of capital letters and punctuation rules may be respected, as in scientific documents for instance, somewhat observed as it is the case of some informative websites, and virtually ignored as it may be in social media.

Although word and sentence segmentation are necessary, actually there is no absolute definition for what each of them constitutes. Both have relatively arbitrary distinctions that dramatically change across the various written systems, though for linguistic computing, there is need for an exact definition of what is required for a later processing; in many cases, language and application define the necessary conventions.

2.2. Tokenization

At present, tokenization is well established and understood in artificial and programming languages. Such artificial languages,

however, may be devised to suppress lexical and structural ambiguities; in the case of natural language, such definition is not the same. In this sort of languages, a word may have several uses and syntax is not strictly defined.

On the one hand, there are languages limited by spaces, as many European languages in which the words' limits are indicated by the insertion of a blank space. On the other, there are also non-segmented languages such as Chinese and Thai, in which some words are consecutively written with no indication of limits between them.

There are three main categories into which the words can be placed, and each category exists in both writing systems: defined by spaces and non-segmented. A language's word morphology may be: isolating, if words do not divide into small units; agglutinative, if words divide into small units called morphemes, with clear limits between them; or inflective, when the limits between morphemes are not clear and component morphemes can express more than one grammatical meaning.

In spite of having a well-formed corpus of sentences, there is a number of issues to solve in tokenization. Much of the ambiguity in tokenization occurs because of punctuation marks, since the same punctuation mark can have different functions in one single sentence.

An initial idea while tokenizing a language limited by spaces may be to consider a any sequence of characters preceded and proceeded by a space as a token; this successfully tokenizes words that are a sequence of alphabetic signs, but it does not consider punctuation marks. In many cases, marks such as commas, semicolons and colons must be dealt with as separate tokens.

Tokenization in non-segmented languages needs a more elaborate approach than a simple lexical analysis.

2.3. Parsing

It is convenient to consider the differing aspects of parsers applied to natural languages and those adapted to programming languages. One of such differences is the power of formalisms in grammar; that is to say, grammar's generative capacity.

Programming languages are usually designed to allow codification by means of a non-ambiguous grammar. To do so, Context-Free Grammar (CFG) subclasses are carefully restricted. On the contrary, natural language needs more powerful instruments; one of the most distinguished cases of expressive power is long-distance dependence, for example, in English, question words.

Another difference has to do with the ambiguous structure of natural language. Commonly, there are grammar rules to apply over a sentence; a classic example is the following: "Put the picture frame in the box on the table". If we assume that "put" categorizes two objects, then there are two possible analyses; "put the picture frame [in the box on the table]" and "put [the picture frame in the box] on the table".

The purpose of a general grammar would be to account for what it is possible in any context. Much of the work in parsing deals with the ways in which enormous potential search spaces may be efficiently treated and how the most appropriate analysis can be chosen.

The third difference bases on the fact that natural-language texts are inherently noisy. Unlike natural language, a computer language

has a complete specific syntax, which means, by definition, that each correct input chain is analyzable. In natural language, it is difficult to distinguish if a fault produced by the result of the analysis comes from an input error or lack of grammar coverage.

2.3.1. Context-Free Grammars

Ever since Chomsky (1956) introduced them, Context-Free Grammars (CFG) have been the most influential grammar formalism to describe a language's syntax. This is because many of the grammar formalisms come from or are somewhat related to CFG.

The standard way to define a CFG, on the one side, is by means of the tuple $G = (E, N, S, R)$, where E and N are separate finite sets with terminal and nonterminal symbols, respectively, and the set $V = E \cup N$ contains grammar symbols. Furthermore, R is a finite set of production rules in the form of $A \to \alpha$, where $A \in N$ is a nonterminal and $\alpha \in V$ is a sequence of symbols.

Capital and small letters are utilized to identify nonterminal and terminal symbols, respectively.

A grammar is in the Chomsky Normal Form (CNF) if every rule is a terminal unitary rule in the form of $A \to w$, or a nonterminal binary rule in the form of $A \to BC$. It is always possible to transform a grammar into the CNF that accepts the same language; however, the transformation can radically change the grammar structure. For example, if the original grammar has n rules, the transformed one may have, in the worst of cases, $O(n2)$ rules.

This normal form can be attenuated by allowing nonterminal unitary rules in the form of $A \rightarrow B$. The transformation for this form is simpler and the transformed grammar is structurally similar.

2.3.2. Parse trees

The standard way to represent the syntactic structures of a grammar sentence is by means of a parse tree (figure II.1), which is a representation of all the steps to derive a sentence from its root node. This means that each internal node in the tree represents the application of a grammar rule.

In practice, CFG are not widely used to set up grammars. One of the reasons is that CFG are not sufficiently expressive; i.e., they do not describe all the peculiarities of natural language, but the main reason is that they are very difficult to use.

In parsing, various procedures take place, a recognizer is a function that ascertains if an input sentence agrees with the established grammar. Likewise, a parser is a recognizer that produces structural analyses in accordance with grammar. A robust analyzer produces both outputs and partial analyses, even if the input is not fully covered by grammar.

We may think that a grammar induces a search space that comprises a set of states representing stages of rewriting of successive rules, and a set of transitions between those states. When a sentence is analyzed, the analyzer must rewrite the grammar rules in a certain sequence. A sequence that connects state S, the chain that comprises the starting grammar category, and the state that comprises the exact chain produced by input words is called a derivation. Each state in the

sequence is then a V state, and it is called sentential form. If such a sequence exists, it is said that the sentence complies with grammar.

Parsers may be classified into several dimensions according to the way derivations are made; one of such dimensions refers to the invocation rule. These may be up-bottom, in which every sentential form is produced by its predecessor by means of replacing a nonterminal symbol A with a chain of terminal or nonterminal symbols $X_1...X_d$, where $A \rightarrow X_1...X_d$ is a grammar rule. Inversely, in bottom-up derivation, every sentential form is produced by replacing $X_1...X_d$ with A, given the same grammar rule, and successively applying grammar rules in the opposite direction.

Another dimension refers to the way in which the analyzer deals with ambiguity; particularly, if the process is deterministic or not. For the first case, an irrevocable option may only be taken if the analyzer faces local ambiguity. This election is typically based on some way to look ahead or some systemic preference. A third dimension refers to if the analyzer goes through the input from the left or right or in any other order.

2.3.3. Flexibility

In a parser, it is logical to expect that inputs correspond to chains in the formal language L(G) produced by means of G grammar. However, a natural language parser will always be exposed to some inputs that are not in L(G). A source of this problem is underproduction, which is caused by a lack of coverage of G regarding natural language; another problem is that inputs may contain errors, that is to say, it may be poorly formed. Albeit, regardless of the reasons why the inputs are

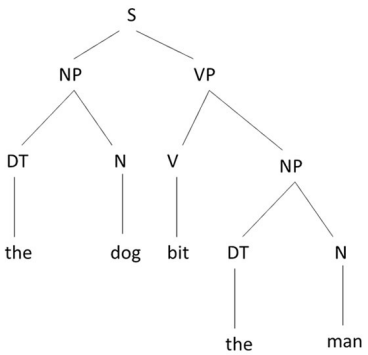


Figure II.1: Example of a parse tree

not in L(G), it is usually advisable to try to recover as much relevant information as possible, instead of delivering no results whatsoever. In a sense, robust analysis means that small deviations from the expected input will only cause small inconveniences in the analysis results, while extreme deviations may produce large inconveniences.

It is plain to see that a robust system requires methods that somewhat compromise the traditional idea of recovering a full and accurate analysis; in order to avoid a situation in which the parser can only stop and report a fail in the analysis of an input, one option is to relax some of the grammar restrictions up to the point that a potential non-grammatical sentence obtains a full analysis.

The key problem of this approach is that the number of alternatives to relax the restrictions that are compatible with the analyses of the entry may grow out of control as the number of errors increase and the search for the best solution is, therefore, very difficult to control.

One might then focus on grammar design, making it less rigid with a view to allowing it to process with more tolerance. The amount

of information contained in the structural representations produced by the parser usually relates to the distinction between in-depth and cursory analyses. An in-depth analysis typically accounts for long-term dependencies; conversely, a cursory analysis makes use of a more schematic representation. This is carried out by means of assigning all the possible grammar and syntax tags to every word, in the first place; then pattern recognition rules are applied to perform disambiguation on the tags and as a result to decrease the number of analyses.

The result is a structure of dependence in the sense that it only offers relations between words and it may be ambiguous because the depending entities are not fully specified. A second option is to sacrifice full inputs, i.e., the input coverage, which entails only letting grammatically well-formed fragments pass.

A third option is to sacrifice the traditional notion of a constructive analysis; i.e., sentence analysis by means of constructing syntax representations defined by grammar rules. Instead, being able to use a process of elimination, which consists in beginning with the most conditions and gradually reducing the analyses that are illegal according to the restrictions set, up to having only valid analyses.

2.3.4. Disambiguation

A basic observation is that despite a general grammar allows for a large number of analyses of almost any nontrivial sentence, many of such analyses will be virtually impossible in the context of a particular domain.

Based on a sufficiently extensive corpus analyzed by the original grammar, it is possible then to identify common combinations of

grammar rules and save them as a single set of rules. The result will be a specialized grammar, which in comparison with the original grammar will have a larger number of rules, but a simpler structure, reducing ambiguity and allowing fast processing following a left-right analysis.

Another possibility is to resort to a hybrid method that hierarchizes a set of analyses on the basis of their probability in a certain domain and supported on data from a training dataset. Consequently, disambiguation allows, naturally in one way or another, applying statistical inference.

2.4. Semantic analysis

For numerous texts, NLP specific applications for semantic analysis may include information retrieval, information extraction, summary generation, data mining and automatic translation, among others. Semantic analysis is also very relevant in the endeavor of producing web ontologies and knowledge representation systems.

Likewise, there is a great value in carrying out semantic analyses as correctly as possible, for they reflect the speakers' cognitive reality. This makes it easier to model the native speakers' intuition and simulate their inference processes, though there is uncertainty as to which extent this is possible. In general linguistics, semantic analysis refers to the organization of the meaning of words, set expressions and sentence content. In practice, this means to turn the original expressions into some sort of semantic language.

Many approaches influenced by psychological logic have focused on the conditional meaning of the truth, though such analyses are

severely limited to allow fully grasping the use of ordinary language or to support the practical applications required, especially those that entail a human-computer interface or ingenious reasoning by regular users. There exists a natural division between lexical semantics, which deals with the meaning of words and the fixed combinations of them, and supra-lexical semantics, which has to do with the meanings of the largely undefined number of combinations of words and sentences allowed by grammar.

Presently, many a linguist speak of lexicogrammar, instead of keeping a strict connection between lexicon and grammar. This is partly because it is evident that the combinatory potential of words is heavily defined by their meanings; to a certain extent, it is clear that many grammar constructions have construction-specific meanings.

An important point in sematic theory refers to the possibility of drawing a well-defined line between semantic content, in the sense of content codified in the lexicogrammar and the general encyclopedic knowledge. By and large, the earliest evidence of linguistic semantics comes from the native speakers' interpretations of the use of linguistic expressions in context, from the natural observation of language in use and from the distribution of linguistic expressions; this last refers to use, collocation and frequency patterns.

A frequently identified requirement in NLP semantic analysis is to resolve ambiguity; from the standpoint of the computational processing of texts, many of the sentences produced by humans are open to several interpretations, since words can have more than one meaning (lexical ambiguity) or because certain words such as quantifiers, adverbs of manner or negative operators may provide a sentence with a

non-deterministic meaning (scope ambiguity) or the intended reference for pronouns is not clear (reference ambiguity).

As regards lexical ambiguities, it is common to distinguish between homonymy and polysemy; the former is the similarity relation in the written form or pronunciation; that is to say, two words with a different meaning, while the latter refers to a word with more than a meaning.

Other problems for lexical semantics come from the existence of figures of speech or multiword units; the latter are words whose meanings are not predictable via the individual words' meanings.

2.4.1. Theories and approaches to semantic representation

A number of theories and approaches to semantic representation may be placed into two categories: formal *v.* cognitive and compositional *v.* lexical.

Formal theories have been strongly defined as of the end of the 1960's, while cognitive approaches have become popular over the last three decades, from the influence of cognitive science and psychology.

On the one side, compositional semantics has to do with the construction of meaning, in ascending order, starting with lexical components; on the other, lexical semantics has as a goal to analyze the meaning of lexical components through their internal structure and content, or through the representation of relations with other lexicon elements.

2.4.2. Logical approaches

The logical approaches of meaning generally focus on compositional problems, from the hypothesis that the meaning of supralexical

expressions is defined by their components' meanings and the ways these expressions combine.

Various logical systems have been and are being developed in the field of linguistic semantics and NLP. One of such systems is *predicate logic*, in which the properties of the set of objects can be expressed by means of predicates, logical connectors and quantifiers. This is carried out furthering syntax and specifying how the elements of logical language may be combined to form correct logical expressions and by providing semantics, specifying what such expressions mean in the context of the logical system.

Even though sometimes it was presented as a general knowledge theory, *predicate logic* is not sufficiently robust to account for the complexities of semantic meaning and it is fundamentally different from human reasoning.

2.4.3. Discourse representation theory

Discourse representation theory was developed by Kamp (1981) in the early 1980's in order to account for semantics in discourses or texts; that is to say, coherent sequences of phrases or sentences. The basic idea is that as a discourse or text develops the listener builds a mental representation, accounted for by the so-called discourse representation structure, and that each input sentence asks for additions to this representation. This way, it is a dynamical approach for natural language semantics.

Discourse representation theory needs the following components: a formal definition for the language representation that comprises a recursive definition of all the well-formed discourse representation

structures and a model – theoretical semantics for this set's members; a procedure for construction that specifies how a discourse representation structure may be added when new information is made available.

The approach of discourse representation theory is well suited to deal, for example, with anaphora resolution.

2.4.3.1. Pustejovsky's generative lexicon

Another dynamical approach to semantics, though focusing on lexical elements, is Pustejovsky (1991) generative lexicon theory.

Pustejovsky puts forward that, in particular contexts, lexical components take on various meanings. He developed the idea of a lexicon in which the words' meaning in context may be flexibly derived on the basis of a multilevel representation that encompasses at least the following levels:

1. Argument structure: a specification of the number and sort of logical arguments and how these syntactically unfold.

2. Event structure: definition of the sort of event of a lexical element and a phrase. The sort of event comprises states, processes and transitions; structuring in sub-events is possible.

3. Structure of subjective qualities: explanation modes, which comprise subjective qualities of four sorts: constitutive, what is the object made of?; formal, what is the object? How is it distinguished in a broad domain?; of defined end, what is the objects' purpose and function?; and of nature, how does the object appear? Factors that influenced its appearance.

4. Structure of lexical inheritance: identification of how a lexical structure is related to other structures in the lexicon and its contribution with the lexicon's global organization.

The generative processes of Pustejovsky's system interconnect the representation levels with a view to providing a compositional interpretation of the lexical elements in context. The generative lexicon has a number of purposes in the logic-inspired approaches for semantics. It is guided by lexical de-compositional approach broken down for linguistic semantics and, at once, it offers tools that allow learning meanings in context by means of a compositional calculation.

2.4.3.2. Natural semantic metalanguage

Natural semantic metalanguage is a de-compositional system based on semantic bonuses empirically established; i.e., simple and undefinable meanings that seem to be present as meanings for the word in all the languages.

This system uses a metalanguage, which essentially is a standardized subset of natural language. It comprises a subset of words with their meanings and another with their associated syntactical properties.

The formal way to represent the meaning in the approach of natural semantic metalanguage is semantic explanation. This is a reductive paraphrase; that is to say, an attempt to express in other words what the speaker is expressing when they pronounce the expression that is explained. Devised by Wierzbicka (1972), natural semantic metalanguage has been developed and refined for 35 years; although

this approach is possibly the best lexical semantics theory, it has few NLP applications.

Researchers working on this approach recognize that for a large number of words in a specific language, it is not possible to produce plausible explanations in terms of unique semantic bonuses. Instead, explanations typically need a combination of semantic bonuses and certain lexical meanings, known as semantic molecules in the theory of natural semantic metalanguage. Semantic molecules work as construction blocks to accomplish more complex concepts. These molecules can also nest, one into another, creating semantic dependence chains.

2.4.3.3. Object-oriented semantics

Object-oriented semantics is a relatively new field in linguistic semantics. Despite it is quite restricted in semantic application domains, it is mainly applied to the representation of verbal meaning.

The basic motivation that drove the dissemination of the object-oriented computational paradigm in linguistic semantics is its intuitive accessibility. The human cognitive system bases on entities and on what they are, how they relate with other entities, what occurs to them and what they do, and how they interact with others. This relates to the object-oriented approach, in which the concept of object is central, whose characteristics, relations with other states, behavior and interactions are rigorously modeled.

2.4.3.4. Relational facts and ontologies

Semantic relations between lexical elements (meaning relations) produce the base for the networks of words such as the digital database called WordNet (Fellbaum, 1998) and other similar approaches for languages other than English; for example, the multi-language database EuroWordNet (Vossen, 1998, 2001). The use of ontologies and linguistic structures is becoming an outstanding area in knowledge representation and NLP.

Meaning relations can be seen as witnesses of a lexicon's sematic structure. There are vertical and horizontal meaning relations. Horizontal relations include synonymy, i.e., words with the same meaning, but different linguistic form. There are also two main vertical relations that include hyponymy and meronymy. The first occurs when the meaning of a lexical element, the hyponymy, is more specific than the meaning of the other, the hypernymy; the second occurs when the meaning of a lexical element specifies that it is part of other lexical elements; for instance, finger is a meronym for hand.

Ontologies have been developed and utilized in artificial intelligence and knowledge representation, and more generally in computer science.

Chapter 3

Methods to identify deception

3.1. Background

Detecting deceptive text by means of machine learning algorithms tries to identify the characteristics that define these texts according to a previous set of data called training set. In this way, algorithms can analyze a new text (not contained in the training set) and identify if it belongs to any specific class.

Researchers have studied deception detection by means of various sources of characteristics. Some of these methods are based on Bag of Words (BoW) tools, in which the order of words is not considered (context); others add syntactic information as characteristics. In some cases, general indications of deception are sought (DePaulo *et al.*, 2003) such as use of unique words, self-references or modifiers, among others.

By and large, linguistic approaches (such as ns-grams) analyze relations between words; conversely, BoW approaches (LDA, WSM, LIWC, n-grams) overlook grammar and even the order of words and base mainly upon the counting of each words' repetitions.

In order to detect deceptive texts, a commonly used technique is n-grams. This method can extract characteristics from a text on the basis of various elements; for instance, words, syllables, phonemes, letters, among others.

In Fusilier *et al.* (2015), word n-grams were compared with character n-grams. The latter have proved to have better performance in OpSpam (described in section 4.2.2). Although the best n-grams accomplish acceptable results on its own right, they are generally aided by other NLP techniques, since the combination of characteristics consistently improves the results.

Another BoW approach is to resort to Linguistic Inquiry and Word Count (LIWC), which includes a classification of words and a counting tool. For example, Newman *et al.* (2003), while analyzing LIWC word categories, found that liars use fewer self-references and use words of negative emotions. This work set the foundations for LIWC to be widely used by other researchers (Schelleman Offermans and Merckelbach, 2010; Toma and Hancock, 2012).

Hauch *et al.* (2012) introduced a meta-analysis of a number of research works on deceptive text detection; this meta-analysis focused on specific linguistic characteristics, e.g., those contained in LIWC. Their findings suggest that liars use certain linguistic categories at a different rate than individuals who tell the truth.

Deception detection has been applied in various particular situations. In the research by Williams *et al.* (2014), a comparison between lies told by children and by adults was carried out in order to detect deception in at court when children testify. The authors chose 48 children and 28 adults to produce a dataset; half of the children and adults told

lies, and the other two halves told the truth. This way, LIWC was utilized to generate samples for classification. Results showed that there are significant differences between true and false texts, mainly in linguistic variables such as singular self-references (for example, *I*, *my*, *me*), self-references (for example, *we*, *our*, *us*) and negative emotions. Moreover, the research results demonstrate that linguistic variables were found at various proportions, depending on if the lie was told by a child or an adult.

Studies developed by means of BoW tools have been successful; however, in an effort to improve the results, the sentences' context has been considered in the analysis of words' syntactic relations with dependency trees (Xu and Zhao, 2012). Generally, the use of syntactic relations has not had an outstanding performance in classifying deceptive text; even though, complementing this method with a BoW approach may improve results.

In a study by Pérez Rosas and Mihalcea (2014b), the characteristics were generated using several approaches (for instance, parts of speech (POS), context-free grammars (CFG), unigrams LIWC and combinations of them). The authors predicted, with an accuracy between 60 and 70%, if a man or a woman had written a deceptive text. In the results they displayed, the use of POS and CFG did not significantly improve the accuracy in comparison with unigrams and LIWC. This suggests that BoW approaches have a performance similar to linguistic approaches.

In order to ascertain the accuracy of learning algorithms, datasets called corpus/corpora (singular, plural) are resorted to; these can be divided into two subsets: the aforementioned training set, which gives the classifier previous knowledge; and the testing set,

which contains documents that will be classified in accordance with the previous knowledge.

3.2. Detection of deception in controversial topics

Mihalcea and Strapparava (2009) analyzed detection of deceptive and truthful texts by means of machine learning. What they intended was to automatically account for underlying characteristics in the text to allow identifying if a text was deceptive or truthful. The authors solely based on obtaining linguistic characteristics, even though other works include non-linguistic ones (e.g., acoustic), this because the most information on the web is in texts.

The goal of the research was to find out if deceptive and truthful texts are actually distinguishable; that is to say, if there are characteristics that can only be found or are more likely found in deceptive texts. In order to solve this, the authors produced a dataset on controversial topics; which comprises deceptive and truthful short texts.

Specifically, the dataset generated (described in section 4.2.3) has three topics: *abortion*, *capital punishment* and *best friends*. As in any case, and to assess the approach proposed, it was necessary to tag the dataset; i.e., each text has its respective identifier to find out if it is deceptive or truthful. For tagging, the authors opted to use Amazon Mechanical Turk, which is a web tool that allows different users help in return for a monetary contribution.

The dataset was generated by means of two processes. In the first place, in the case of *abortion* and *capital punishment*, users had to write their actual opinion on these topics; then, they were asked to write the

opposite opinion. This way, the result is a text with actual information and another with false. Secondly, about *best friends*, the users were asked to think of their *best friend* and describe the reasons for their friendship; then, they were asked to think of someone they were not able to stand and described them as their *best friend*.

As a result, in the first case, the respondents lied about their beliefs; whereas in the second, respondents lie about their feelings for someone else. Eventually, 100 truthful and 100 deceptive texts were gathered per topic, totaling 600 texts.

With a view to finding out the most relevant characteristics of deceptive and truthful texts, they devised a method to assess to which extent words tend to be one or the other class. Such method bases on the following measures: deceptive text coverage (formula 3.1), which calculates the fraction of the number of tokens of word W_i in the corpus of deceptive texts (D) out of the total tokens in corpus D. In like manner, truthful text coverage (formula 3.2) performs the same calculation, but this time for the corpus of truthful texts (T).

Finally, the dominance value of class C (classes included in LIWC, ver. 2001) is calculated (formula 3.3), which is the ratio between the coverages of truthful and deceptive texts, calculated for the same class.

The three topics were mixed to have two general datasets; T and D, truthful and deceptive texts, respectively. The dominance values of each class C were calculated on these general sets. This way, a result close to 1 will indicate there is a similar distribution of words in class C for both truthful and deceptive texts. Furthermore, if the result is less that 1, it will mean that the predominant class is truthful text; conversely, if the result is above 1, it will mean that the predominant class

is deceptive text. Table III.1 shows some dominant class values obtained by the authors.

In the experiments, texts were classified using a naïve Bayes (NB) classifier and a Support Vector Machine (SVM). Firstly, the three topics were classified separately; table III.2 shows the results as well as the accuracy values obtained by each classifier and the average accuracy of the three topics. It is also noticed that on average the best result was obtained by the NB classifier.

Following Mihalcea and Strapparava (2009), a study was run in order to detect deceptive text written in Spanish: Almela *et al.* (2012) gathered a new dataset with topics ranging from LGBT adoption, bullfighting, and feelings for best friends. One hundred false and one hundred truthful documents were gathered for each topic with an average of 80 words per document. Several LIWC dimensions were utilized to accomplish a more accurate classification by means of a SVM.

$$Coverage_D(C) = \frac{\sum_{W_i \in C} Frequency(W_i)}{Size_D} \tag{3.1}$$

$$Coverage_T(C) = \frac{\sum_{W_i \in C} Frequency(W_i)}{Size_T} \tag{3.2}$$

$$Dominance_D(C) = \frac{Coverage_D(C)}{Coverage_T(C)} \tag{3.3}$$

Table III.1: Dominant LIWC classes of deceptive and truthful text

Class	Score	Example of Words
Deceptive text		
Metaphor	1.71	*god, die, sacred, mercy, sin, dead,hell, soul, lord*
You	1.53	*you, thou*
Other	1.47	*she, her, they, his, them, him, herself, himself, themselves*
Humans	1.31	*person, child, human, baby, man, girl,humans, individual, male, person, adult*
Accurate	1.24	*always, all, very, truly, completely, totally*
Truthful text		
Optimum	1.71	*best, ready, hope, accepts, accept, determined, accepted, won, super*
I	1.53	*I, myself, mine*
Friends	1.47	*friend, companion, body*
Myself	1.31	*Our, myself, mine, ours*
Vision	1.24	*Believe, think, know, see, understand, found, thought, feels, admit*

Table III.2: Accuracy by topic using a 10-fold cross validation

Topic	Naïve Bayes (%)	SVM (%)
Abortion	70.0	67.5
Best friend	67.4	65.9
Capital punishment	75.0	77.0
Average	70.8	70.1

3.3. Detection of deception in opinions about hotels

The goal of the research by Ott *et al.* (2011) was to identify deceptive options, which are defined as fictitious opinions that are deliberately written to seem authentic.

To reach this goal, the authors explored three approaches. The first one was to look at the problem as a categorization task by means of n-grams and classifiers, this with a view to tagging the documents as truthful or deceptive. The second option was a psycholinguistic approach, which intended to find out the psychologic states, e.g., negative emotions, related to deception. Finally, in the third approach the task was tackled as a genre problem by means of which the drafting of deceptive texts is perceived as imaginative, whereas that of truthful texts, as informative.

The classification of 400 deceptive and 400 truthful texts allowed the authors to identify that machine learning classifiers, trained with characteristics obtained from psycholinguistic approaches and genre identification, statistically surpass the level of text-categorization approaches such as n-grams.

Moreover, they remarked that human capacities to differentiate between deceptive and truthful texts, as displayed in table III.3, are limited. This was ratified through the classification of a subset of 160 documents by three participants. The table also includes less simplistic measures than accuracy; namely: precision (P), recall (R) and f-measure (F).

The authors also strongly stressed that text identification has to be undertaken considering the context and motivation that drove deception, not only through the identification of a set of universal signs of deception.

On the one side, they put forward generating characteristics based on grammar tags (POS tags) supported on some research works (Douglas and Broussard, 2000; Rayson *et al.*, 2002) which have verified that frequency and distribution of grammar tags have a certain relation with genre identification. The authors thus built vectors of characteristics based on the frequency of each grammar tag.

On the other, and as a comparison of the results from the approach they proposed, they resorted to LIWC, ver. 2007, generating one characteristic by each of the 80 dimensions included in this dictionary. Finally, as regards the sources of characteristics, the authors used n-grams because they allow capturing the documents' content and context. As well, characteristics were generated from unigrams, bigrams and trigrams.

In order to carry out the classification of documents, they used a NB classifier and an SVM with a 5-fold cross validation. The results are shown in table III.4, in which it is noticed that the combination of LIWC characteristics and bigrams, classified with an SVM, had a better result. Besides, as regards accuracy and F-measure, all the characteristics generated and classified with machine learning surpass human classification (table III.3).

Although the authors propose to identify text by means of a genre-identification approach, in table III.4 it is noticed that the generation of characteristics by means of grammar-tag frequency was not the best option to produce sufficiently informative vectors. As the

authors argue that all characteristic of combinations of sources were tried in the experiments, and since no more POS-approach combinations are displayed, it may be inferred that combining the grammar-tag approach with another was detrimental to the classification.

Table III.3: Accuracy, precision (P), recall (R), y F-measure (F) of man-made deceptive text identification

		Truthful			Deceptive		
Participant	Accuracy	P	R	F	P	R	F
Judge 1	61.9	57.9	87.5	69.7	74.4	36.6	48.7
Judge 2	56.9	53.9	95.0	68.8	78.9	18.8	30.3
Judge 3	53.1	52.3	70.0	59.9	54.7	36.3	43.6

Table III.4: Results of the application of various generation sources of characteristics and application of two classifiers

		Truthful			Deceptive		
Characteristics	Accuracy	P	R	F	P	R	F
Genre identification							
POS/SVM	73.0	75.3	68.5	71.7	71.1	77.5	74.2
Psycholinguistic approach							
LIWC/SVM	75.8	77.2	76.0	76.6	76.4	77.5	76.9
Text categorization							
Unigrams/SVM	88.4	89.9	86.5	88.2	87.0	90.3	88.6
Bigrams/SVM	89.6	90.1	89.0	89.6	89.1	90.3	89.7
LIWC+brigrams/SVM	89.8	89.8	89.8	89.8	89.8	89.8	89.8
Trigrams/SVM	89.0	89.0	89.0	89.0	89.0	89.0	89.0
Unigrams/NB	88.4	92.5	83.5	87.8	85.0	93.3	88.9
Bigrams/NB	88.9	89.8	87.8	88.7	88.0	90.0	89.0
Trigrams/NB	87.6	87.7	87.5	87.6	87.5	87.8	87.6

3.4. Detection of deception in statements

In Fornaciari and Poesio (2012) research, the identification of deceptive text was undertaken from the creation of subsets out of a corpus. The grouping of homogeneous subsets of subjects occurs in gender separation; this is to say, if the subject who wrote the text is a man or a woman. The grouping of sets was carried out both, automatically by means of automatic grouping techniques, and using the metadata in the sets. This way, groups, as the authors contend, may increase efficiency in detecting deceptive text, on the grounds that a similar behavior in the authors' drafting may influence on deception detection.

In this study, a set called DeCour (Deception in Court) was created; it is the set of transcriptions of 35 hearings that took place in four Italian courts. The texts comprise questions for the witnesses testifying in favor the defendant; as a result the dataset contains tags that report whether the transcribed sentence or answer is truthful or deceptive.

The authors created vectors with three sorts of characteristics: the first adds basic information such as sentence size and the number of words with more than six characters; the second adds lexical information by means of LIWC 2001, aggregating the 80 dimensions of such version to the vector of characteristics; the third characteristic adds the n-gram frequencies of lemmas and grammar tags. In the experiments, the authors considered from unigrams up to five-grams.

By means of obtaining n-grams, with n values from 1 to 5, two lists of outstanding n-grams were made; one corresponding to deceptive text and the other to truthful text. These listings were compared to

prevent one same characteristic from appearing in both; this way, only particular information that enables efficient classification is generated. Three experiments were carried out using the dataset created. The first experiment consisted in using the entire dataset by establishing training and testing sets.

For the second experiment, subsets of the entire corpus were created in such manner that vectors that were too far from the groups (*outliers*) were suppressed. These groups were established by means of a non-supervised pattern grouping method.

In the third experiment, the metadata collected in the corpus were used. At first, this had the following information: gender (male or female), birth place and age at the time of declaring. Finally, the authors decided to consider only the subjects' gender, specifically, only male respondents. This restriction heavily impacted on the training set, but not to such extent on the testing one.

The testing set comprises 426 sentences, of which 190 are deceptive and 236 truthful; the classification results of the three experiments are shown in table III.5. It is noticed, on the one hand, that the worst result, regarding the F-measure average, is obtained when only men are considered; this may come from the drastic reduction of the training set. On the other, the best result was obtained in the experiment in which the *outliers* were suppressed, which allowed a more homogeneous group.

3.5. Detection of deception in combinations of domains

In section 3.4, it was considered that the creation of subsets on the basis of gender and outliers would have a certain impact on the

Table III.5: Results of the classification of corpus DeCours, showing: precision (P), recall (R), F-measure (F) and the mean of F-measures as well

		Truthful			Deceptive		
Experiment	Mean F	P	R	F	P	R	F
Entire corpus	60.1	62.9	94.0	75.3	80.8	31.0	44.9
No outliers	64.0	66.8	93.8	77.9	81.0	36.2	50.0
Males	59.5	67.8	94.2	78.9	74.4	27.4	40.0

classification efficiency. In like manner, in the research by Pérez Rosas and Mihalcea (2014a), experiments are run with texts written by people from various countries in order to verify to what extent it is possible to identify deception, for example, using texts from one country to identify deception in those from other countries.

Most of the studies on deception detection focus on documents from a certain country and language, mainly in English. In this regard, the authors argue that texts mainly in web sites tend, at a certain proportion, to be written by people who have different language and culture; this implies that people have various beliefs and moral values. Therefore, it is contested if the approaches based on a single language and culture might be equally applicable to a mix of intercultural texts and even in a different language (using translation tools).

In this study, deception detection was carried out with texts from three different cultures: United States, Mexico and India. Owing to this, the authors generated three datasets that approach, for each language, three controversial topics: *abortion*, *capital punishment* and *best friends*. The above is as in the study presented in section 3.2, save for texts

gathered in Mexico, which were obtained from a website created by the authors due to the scarce participation in Amazon Mechanical Turk.

The sources of generation of characteristics for this study were two. As in most of the research works shown in this section (state of the art), unigrams are required to generate a part of the vector of characteristics, so it may be inferred that words on their own are an important informative source to detect deception.

To complement the vector of characteristics, a second tool that has become very popular in deception detection was used; this is LIWC, which as stated before is an approach that provides psycholinguistic analyses.

Table III.6: Experiments on each country's corpora, separately

Topic	LIWC	Unigrams	
		In domain	Crossed domain
English: USA			
Abortion	73.0	63.8	80.4
Best friend	73.0	74.5	60.8
Capital punishment	58.1	58.1	77.2
English: India			
Abortion	56.0	46.0	50.0
Best friend	71.4	60.5	57.2
Capital punishment	63.5	57.5	54.0
Spanish: Mexico			
Abortion	62.2	52.5	57.7
Best friend	75.3	66.7	50.5
Capital punishment	62.2	54.9	63.4

Two sort of experiments were run: one is a classification of the datasets with documents from the same country; the other entails a crossing of domains; i.e., taking two of the three available domains to produce the training set and use the third domain to produce the testing

Table III.7: Culture classification crossed with LIWC and unigrams as sources of characteristics

Topic	LIWC	Unigrams
Training: English: USA, testing: English: India		
Abortion	52.3	57.9
Best friend	59.5	51.0
Capital punishment	53.5	59.0
Training: English: India, testing: English: USA		
Abortion	62.5	55.5
Best friend	55.8	53.2
Capital punishment	39.2	50.7
Training: English: USA, testing: Spanish: Mexico		
Abortion	53.9	61.5
Best friend	67.7	65.0
Capital punishment	62.2	59.8
Training: English: India, testing: Spanish: Mexico		
Abortion	43.6	55.1
Best friend	60.8	67.2
Capital punishment	59.8	51.2

set; this experiment was carried out both with mixed topics, in which the crossed domain is applied on each culture separately, and with crossed cultures, in which the crossed domain is applied on the three cultures; the experiment goal was to try whether the characteristics that work to detect deceptive text in a certain country's language are equivalent or useful to look for deception in the language of another country (with a different culture).

Table III.6 shows the results of the classification, individually, for each culture, by means of the generation of characteristics with psycholinguistic approach and using unigrams. On this last one are applied domain classification, in which each topic is classified

separately (*abortion*, *best friend* and *capital punishment*), and crossed domain classification, in which two topics are used for training and the third, for testing.

Table III.7 shows the results of crossed-domain classification using two languages as a training set and the other language as a testing set.

Table III.8 shows the prevailing classes both in deceptive and truthful texts. Plus, some examples of words included into each class. Similar to section 3.2, a score significantly below one indicates that deceptive text is predominant in the class.

3.6. Detection of deception in opinions on books

Deceptive opinions in book sales, according to the study by Fornaciari and Poesio (2014), occur when the authors write a bright opinion about their own books.

With a view to analyzing deceptive opinions in book sales on Amazon, the authors created a new dataset they tagged on the basis of deception signals; their presence defined if the opinion was deemed deceptive or truthful. The approach used for tagging was crowd learning proposed by Raykar *et al.* (2010); besides, the effectivity of various tagging methods was assessed according to the performance of the models generated to detect deceptive opinions.

The authors claim that the largest disadvantage of other works that deal with deception detection is that they lack a dataset with actual texts; conversely, they are artificially created, for example, when respondents are asked to lie about an actual opinion.

Table III.8: Prevailing LIWC classes in deceptive and truthful texts

Class	Score	Example	Class	Score	Example
English: USA					
Deceptive			Truthful		
Metaphor	1.77	*Die, died, hell, sin, lord*	Vision	0.68	*Accept, believe, understand*
Other	1.46	*He, her, herself, him*	I	0.66	*I, me, my, myself*
You	1.41	*Thou, you*	Optimism	0.65	*Accept, hope, top, best*
Other references	1.18	*He, her, herself, him*	We	0.55	*Our, ourselves, us, we*
Negative emotions	1.18	*Afraid, agony, awful, bad*	Friends	0.46	*Buddies, friend*
English: India					
Deceptive			Truthful		
Denial	1.49	*Cannot, neither, no, none*	Past	0.78	*Happened, helped, liked, listened*
Physical	1.46	*Heart, ill, love, loved*	I	0.66	*I, me, mine, my*
Future	1.42	*Be, may, might, will*	Optimism	0.65	*Accept, accepts, best, bold*
Other	1.17	*He, she, himself, herself*	We	0.55	*Our, ourselves, us, we*
Humans	1.08	*Adult, baby, children, human*	Friends	0.46	*Buddies, companion, friend, pal*
Spanish: Mexico					
Deceptive			Truthful		
Certainty	1.47	Jamás(*never*), siempre(*always*)	Optimism	0.66	Aceptar(*accept*), animar(*cheer*)
Humans	1.28	Bebé(*baby*), persona(*person*)	Oneself	0.65	Conmigo(*me*), tengo(*have*), soy(*am*)
You	1.26	Eres(*are*), estás(*be*), su(*his/her*)	We	0.58	Estamos(*are*), somos(*be*), tenemos(*have*)
Denial	1.25	Jamás(*never*), tampoco(*neither*)	Friends	0.37	Amigo(*friend*), amistad(*friendship*)
Other	1.22	Es(*is*), está(*are*), otro(*other*)	Past	0.32	Compartimos(*share*), vivimos(*lived*)

The point is that, by giving someone the chance to lie, it is possible that the characteristics of such deception do not reflect in the text or do it partially; it is even possible that what is being captured is not the same phenomenon. This way, the identification and use of the model will not be congruent with deceptive texts in actual cases.

Due to the above, the authors proposed a new method to identify deceptive opinions in Amazon. This system is based on two principles defined below:

In the first one, a collaboration with an expert on detection of deceptive opinions took place in order to identify a series of criteria to find non-genuine opinions. Some of these opinions were known to be deceptive beforehand, since it was confessed by the people engaged in writing them.

In the second principle, the authors developed an approach that identified the veracity of opinions using indicators of veracity. Finally, they used the algorithm proposed by Raykar *et al.* (2010) to assign each opinion in the corpus to a class.

After creating the corpus, the authors identified a set of signals, whose presence suggested that the texts were deceptive. The signals identified were: 1) Suspicious books. According to the research, these are books that surely or very possibly received deceptive opinions; 2) Time. Sandra Parker, a writer of deceptive opinions, said that the agencies she worked for gave her a 48-hour deadline to write the reviews; so, she suggested paying close attention to books reviewed within short periods; 3) Account. Amazon allows creating user accounts using the users' real names; therefore, it is less likely that users with a real name wrote deceptive opinions; 4) Purchased book. Other piece of information that may be obtained from Amazon is if the book was purchased or not in the store; therefore, the absence of a purchase was considered a signal of deception.

Two experiments were carried out: in the first, the assignation of classes was carried out on the basis of a voting on signals of deception.

Opinions with up to two deception signals were considered truthful, and those with more than two were considered deceptive. In the second experiment, the crowd learning method was used.

In order to make the classification, a vector of characteristics was created by each opinion. Such characteristics were based on unigrams, bigrams and trigrams of lemmas and grammar tags. Then, two lists of the frequencies of each list of sources of generation of characteristics were generated; one for deceptive opinions and other for truthful opinions. With this, the most frequent characteristics of each class were sought. To do so, only the n-grams that appeared more than 300 times in each frequency list were taken into account.

Table III.9 shows the classification of dataset DeRev by means of a support vector machine; it also shows the comparison between tagging methods, with a better performance, though not significant, of the algorithm by Raykar *et al.* (2010).

Table III.9: Classification of DeRev that shows the performance of the tagging methods

Method	Accuracy	Deceptive		
		P	R	F
Class vote	75.4	83.3	63.6	72.1
Algorithm by Raykar *et al.* (2010)	76.3	78.7	72.0	75.2

Chapter 4

A new approach to deception detection

4.1. Sources of generation of characteristics

Three different sources of generation of characteristics (LIWC, ns-grams, WSM), aimed at comparing and combining the method proposed (LDA), were specifically researched in this study.

Most of the research, presented as the current norms (section 3), resorted to unigrams as a basis to add new characteristics with a view to obtaining better performance. Conversely, for the present study, a word space model (WSM) was chosen (section 4.1.1), since it had a similar performance, in addition to being a simpler representation.

Since deception involves cognitive processes, other source of characteristics selected was the LIWC dictionary (section 4.1.3). Since this tool was created to approach psychological studies, it has been utilized in deception detection to generate psycholinguistic information. LIWC comprises a set of words with semantic relations.

However, the main inconvenience of LIWC is the set of pre-established words; therefore, in order to be able to compare LIWC performance as regards a tool that also produces groups of words with semantic

relationships, a continuous semantic space model (LDA, section 4.1.4) was decided upon. This method automatically creates a set of topics and each of them comprises a set of words with certain semantic relationships.

Unlike LIWC, the set of words in LDA may change depending on the dataset to be processed, which suggests that the categories generated are specific for the collection of documents; consequently, the characteristics can be more informative.

In the methods described above, only lexical information, including lexical semantics, is considered. Therefore, considering that various works show that syntactic information offers valuable information, an approach based on the text style (ns-grams) has been explored (section 4.1.2); in comparison with other methods (BOW), ns-grams can consider the context of words in sentences.

4.1.1. Term-document matrix (TDM)

In many works, vectors of words are mostly chosen because they generate relevant characteristics that help classify texts. Owing to this, it was decided to analyze the performance of the characteristics based on a matrix of words, more specifically, using a binary representation.

In order to obtain a single representation, a list of all the words $W_1, W_2, ..., W_n$ in the dataset was made. Then, each document was analyzed looking for the existence of W_n in any of the texts; if it is the case, the characteristic n (F_n) is established at 1; on the contrary, it is established at 0. Figure IV.1 shows the representations of the vectors of characteristics.

4.1.2. Syntactic n-grams

Syntactic n-grams (Sidorov *et al.*, 2014) (ns-grams) are a relatively new characteristic that appeared after considering certain inconveniences of conventional *n*-grams, being their main disadvantage that they do not correctly account for long-distance relations, which produces the effect in which conventional *n*-grams seem to be generated excessively at random.

The main advantage of n-grams is they can be considered a linguistic phenomenon, as they are extracted from a syntactic structure; which does not occur for conventional *n*-grams, arbitrarily generated.

The disadvantage of being able to generate *n*-grams with syntactic information is that a previous syntactic processing is needed; which requires additional resources such as a parser for the specific language in use. Such resources are not always available for all languages.

In the same way as conventional *n*-grams, ns-grams can be generated from a number of elements; that is to say, words, grammar tags or syntactic relations.

In order to assess the performance of ns-grams, in the study by Sidorov *et al.* (2014), a dataset with 39 documents written in English by three different authors was classified. Furthermore, 60% of the documents was used as a training set and the remaining 40% as testing set. A comparison of the performance obtained by the authors is shown in tables IV.1 and IV.2; in these tables, the file size defines the threshold of the *n*-grams considered the most frequent; NA means the threshold defined was not attained. We notice that both syntactic bigrams and trigrams manage to correctly classify all documents.

On the basis of the authors' good results, ns-grams were chosen as source of generation of characteristics in order to find out their efficiency in detecting deceptive text. For such end, the procedure is described below.

To obtain the ns-grams, a syntax tree was built to gather syntactic relations represented by the tree's edges; which are those that join words with the correct tag.

Table IV.1: Comparison between syntactic and conventional bigrams in authorship identification

	Characteristics			
File size	Syntactic relation ns-grams	POS tags n-grams	Character n-grams	Word n-grams
400	100%	90%	90%	86%
1 000	100%	95%	95%	86%
4 000	100%	NA	90%	86%
7 000	100%	NA	NA	86%
11 000	100%	NA	NA	89%

Table IV.2: Comparison between syntactic and conventional trigrams in authorship detection

	Characteristics			
File size	Syntactic relation ns-grams	POS tags n-grams	Character n-grams	Word n-grams
400	100%	90%	76%	81%
1 000	100%	90%	86%	71%
4 000	100%	100%	95%	95%
7 000	100%	100%	90%	90%
11 000	100%	95%	100%	90%

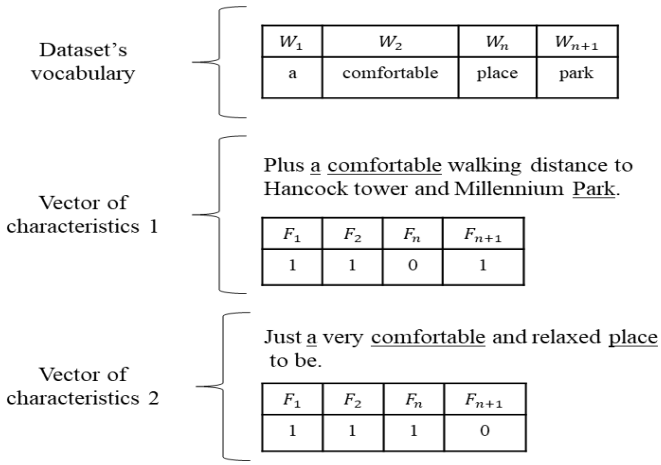


Figure IV.1: Example of vector construction using a model of space of words

Marneffe *et al.* (2006) analyzer was utilized to generate syntactic relations that correspond to the analyzed documents; no collapsed representation was used. An example of the Stanford representation of syntactic relations is presented in step 1 in figure IV.2. In step 2, useful relations are selected to obtain bigrams; it is noticed that prep_to (give-3, kid-7) is related to det (kid-7, the-6) because kid-7 links them both. This way, bigram prep_to-det is obtained.

The documents can be represented by their syntactical relations by means of a vector of characteristics. This was carried out as shown in the previous section, see figure IV.1, but this time replacing words with syntactic n-grams.

4.1.3. Linguistic Inquiry and Word Count

Linguistic Inquiry and Word Count (LIWC) is a tool based on a word dictionary (Pennebaker *et al.*, 2015). This LIWC dictionary, contains

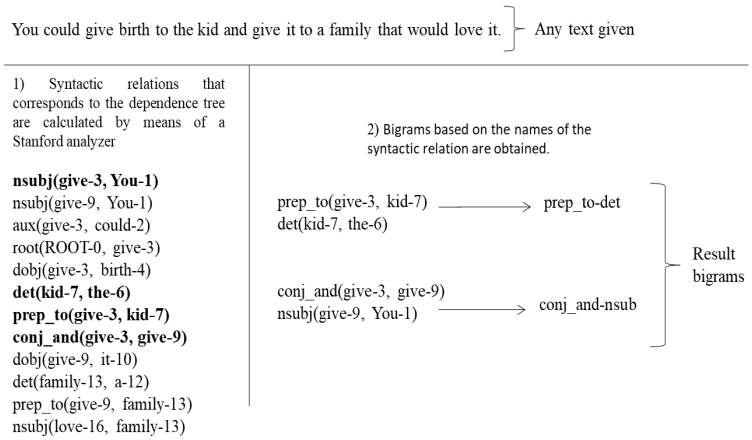


Figure IV.2: Process to generate syntactic bigrams from a sentence

groups of words tagged by humans and was initially used in works related to psychological analysis.

The applications of this resource have recently grown; for example, LIWC has been used in computational linguistics as a source of characteristics to identify authorship (Mohtasseb *et al.*, 2009) and deception detection (Masip *et al.*, 2012), among other applications.

In this work, LIWC version 2007 was used. This version comprises about 4 500 words grouped in 80 categories or dimensions. Figure IV.3 shows an instance of some groups of word contained in LIWC.

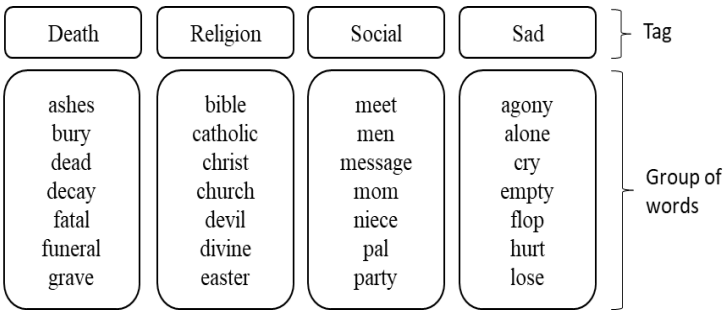


Figure IV.3: Example of some groups of words in LIWC with their tag. A group with only seven words per tag is presented

4.1.4. Latent Dirichlet Allocation (LDA)–continuous semantic space model

Latent Dirichlet Allocation (LDA) is a generative probabilistic method (Blei *et al.*, 2003) for discrete data collections such as text collections.

LDA represents the documents as a mix of various topics; each of them has a set of words which keep some semantic relation between them. Words, for their part, are chosen in function of a probability. The process to select topics and words to generate a document or a set of documents is repeated; as a result, each document generated comprises several topics.

In a simplified way, LDA generative process entails the following steps:

1. Finding the N words that a document will contain according to a Poisson distribution.

2. The election of a mix of topics for the document according to Dirichlet distribution regarding a fixed set of K topics.

3. Generating each word in the document by means of:

- Choosing a topic
- Using such topic to generate the word.

Resorting to a generative model in an inverse manner, LDA analyzes the set of documents to find the most probable set of topics possibly dealt with in a document.

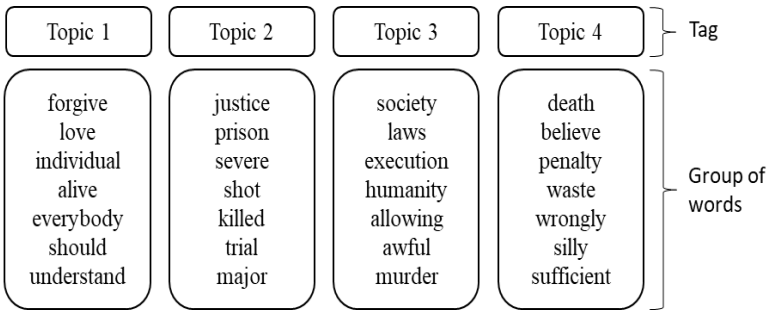


Figure IV.4: Example of topic generation, from texts on *capital punishment*, by means of LDA

We may consider LDA as a tool that generates groups of similar words, as LIWC, but unlike this one, LDA automatically generates the groups of words (topics). Moreover, the groups of words in LDA are not tagged and their content is different depending on the training corpus. For examples of words obtained using LDA, see figure IV.4.

Figure IV.5 shows a document processed by means of LDA (before binarization). Each topic displayed has a particular probability to exist in the document at hand. This way, a high probability of topic (x) in document (y) indicates that x is more likely to appear in y. Each document in the collection has a different distribution of a fixed amount of topics.

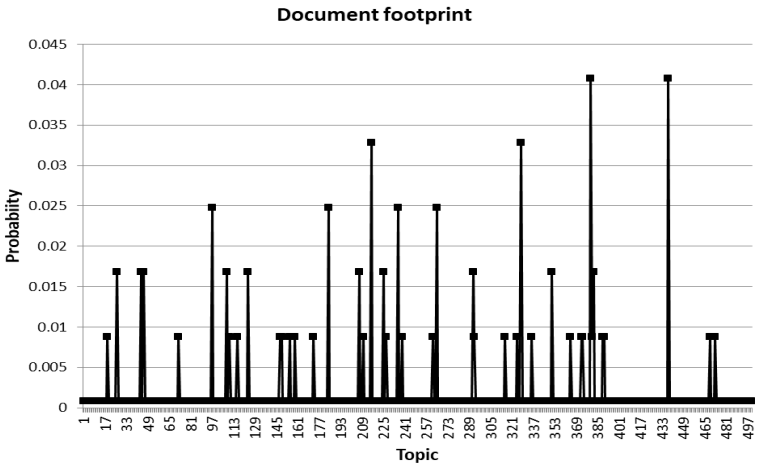


Figure IV.5: Example of a document processed by LDA. The distribution of topics is shown

4.2. Datasets

To assess the method proposed, three different datasets were processed: DeRev (DEception in REViews) (Fornaciari and Poesio, 2014), corpus OpSpam (Opinion SPAM) (Ott *et al.*, 2011, 2013) and a corpus created with opinions on three controversial topics (Pérez Rosas and Mihalcea, 2014a).

The first corpus was collected using penalized deception, while the other two were collected by means of unpenalized deception (Gokhman *et al.*, 2012). The details on these sorts of deception are described in subsequent sections.

4.2.1. DeRev dataset

Dataset DeRev is a corpus of deceptive and truthful opinions from an Amazon webpage. This corpus comprises opinions on books. It is a gold standard that holds 236 texts, out of which 118 are truthful and 118 are deceptive.

In order to reach a high degree of confidence in a correctly tagged collection of deceptive texts, the authors of DeRev considered two research works. The first, by Sandra Parker, was published in Money Talks News; while the second, by David Streitfeld, appeared in *The New York Times*. Parker, on the one side, stated she received a payment to write opinions on 22 books; Streitfeld, on the other, discloses four books whose authors admitted to paying for deceptive reviews. From these facts, an analysis was run, and selection focused on 20 writers of false opinions, producing a corpus with 96 deceptive opinions.

Finally, to obtain the 118 truthful texts, the authors took certain aspects into account to ensure high probability for the section to be correct. The text must not bear any deception indication. Owing to this, selection mainly centered on aspects such as if the opinions were written by users who had utilized their real names and if the opinions were written by users who had bought the book on Amazon, among others.

In this dataset, deceptive and truthful texts were not obtained deliberately, that is to say, the respondents were not asked to lie; conversely, the texts were obtained after the respondent had lied. Owing to this, it is said that the way to build this text was by

means of unpenalized deception; below one deceptive text and another truthful are shown.

Example of deceptive text: *Circle of Lies, by Douglas Alan is a fast paced, gritty crime thriller that introduces us to John Delaney, ex-cop turned lawyer who finds himself defending his best friend against embezzlement and murder charges when no one else will believe him. Alan, a retired trial lawyer, gives us a rather unique look at the "men behind the curtains" of the American justice system with this fast paced, exciting read. Highly recommended!*

Example of truthful text: *This book is well written and very funny, it is a good "guys"; kind of book, and it is an easy read, but don't let that fool you. The story is good and it is in the painful details. I felt like I was working right there with them. However, it is crude and not for the casual reader. I try to read both classic and non-classic writers of all eras, and I am always looking for something weird, funny, and cool to mix it up and this was it.*

4.2.2. OpSpam dataset

Dataset OpSpam is a corpus composed of deceptive and truthful opinions. These opinions are about a number of hotels.

To generate deceptive opinions the creators of corpus OpSpam used Amazon Mechanical Turk (AMT). The authors showed each Turker (worker for AMT) the hotel name and their respective website to complete the task. By using AMT website, Turkers were asked: to imagine they worked in a hotel and the manager asked them to write an opinion of the place, as if they were hosts. The opinion had to seem real and underscore the positive aspects of the hotel. The opinions were limited

to one by Turker to prevent the same person from writing more than one. Likewise, only people living in the US were enabled to undertake the task, and the Turkers had to accomplish an approval grade of at least 90%.

On the one side, Turkers had up to 30 minutes to write the opinion and were paid one USD per accepted opinion. By means of this, OpSpam's authors managed to gather 400 deceptive texts. For their part, truthful opinions were collected from the website of TripAdvisor. A total 6 977 opinions were retrieved from the 20 most popular hotels; then the authors suppressed 3 130 that did not have five stars, 41 that were not written in English, 75 with fewer than 150 characters, and 1 607 that had been posted by people doing this for the first time in Amazon. Finally, OpSpam's authors selected 400 of the rest of the texts.

A dataset composed of 800 texts in total was produced. To build this corpus, the participants were asked to write lies to obtain deceptive text. Consequently, the corpus was produced by means of penalized deception. Below, one deceptive text and another truthful contained in this corpus are displayed:

Example of deceptive text: *I had a wonderful time at the James Hotel while on business in Chicago. The rooms are modern, tasteful and well-kept, while the staff was responsive and efficient. This was the perfect place to unwind after working all day, but also provided an ideal atmosphere to do some more work.*

Example of truthful text: *Despite what other are saying, this was one, if not the best Hotel stay in Chicago I have had. I travel to the Big City about three times a year for pleasure and The James rates up with*

the best. I was upgraded and the staff made the stay worth my special weekend visit. I don't believe you will be disappointed.

4.2.3. Dataset of controversial topics

It is a corpus composed of 600 opinions on three controversial topics, namely: *abortion* (200), *capital punishment* (200) and *best friend* (200). The corpus contains 100 deceptive texts and 100 truthful for each topic. The gathering of texts was carried out over AMT and it was restricted to Turkers living in the United States.

To gather truthful texts, the authors asked the participants their actual opinion on each of the topics. Then, they were asked to lie about such opinions; obtaining thus the deceptive texts. The method utilized to gather this corpus was penalized deception.

Although this corpus also contains texts in English and in Spanish from India and Mexico, respectively, such texts were not used in the present work.

Example of deceptive text: *Abortion is despicable because we're giving women the choice to kill life. Where is the unborn child's voice? Do we not care about those who can't defend themselves? Abortion is systemic murder that needs to be stopped.*

Example of truthful text: *Abortion is a choice. It should be made by both parties if possible, but the woman should have final say if there is disagreement. I think that with counseling, and all the other options available, it is not morally wrong for a woman to choose to have an abortion. A fetus is not viable outside the womb before 26 weeks, and then only barely, and with very expensive care.*

4.2.4. Dataset analysis

With a view to gaining insight into the corpora, table IV.3 shows a summary of the *tokens* and sorts of various datasets previously described; *tokens* show the total number of words in the documents, while types represent the total number of unrepeated words found in the documents. Void words (words with no meaning on their own) were retained in all the experiments.

Table IV.3: Amount of types and *tokens* in the corpora. DTA= Document token average

Corpus	No. of documents	*Tokens*	Types	DTA
OpSpam	800	96 793	6 469	121
DeRev	236	29 990	5 162	127
Abortion	200	15 958	1 997	80
Best Friend	200	11 717	1 718	59
Capital punishment	200	15 615	2 034	78

Table IV.4: Average of types shared by the corpora

Corpus	Average of shared types		
	Deceptive+truthful	Deceptive	Truthful
OpSpam	19	19	19
DeRev	14	16	13
Abortion	14	12	16
Best Friend	9	8	12
Capital punishment	14	13	15

Table IV.4 shows the average of types regarding each corpus; i.e., the average of unrepeated words that the documents in the corpus have in common. The results are shown for all the documents (deceptive +

truthful) and for separate classes. The results displayed in this table also indicate that biases in the sets are less likely to occur, since the average of shared words is very close between united and separate classes, which suggests that there are not too many prevailing words by class. On the contrary, did they exist, we infer that the number of shared words would have noticeably decreased when classes were combined, due to the fact that they would have few words shared in common.

4.3. Construction of vectors of characteristics

Various experiments were run using various sources (described in section 4.1) and combinations of them to find the best configuration of characteristics for deception detection (according to the datasets analyzed in this work). For all the experiments, classification was performed with the machine learning algorithm Multinomial Naïve Bayes implemented under WEKA (Hall, 2009) with five-fold cross validation.

In an attempt to generate a model to best represent deception detection, a selection of attributes was set up so that repetitive and irrelevant characteristics were suppressed.

In some cases, the selection of attributes (Guyon and Elisseeff, 2003) helps obtain a model to detect deception with higher efficiency. It also reduces the dimensions of the vectors of characteristics; for example, in the present study, the vectors composed of 4 000 characteristics were reduced to about 60 characteristics.

In the attribute selection process, WEKA was utilized with an evaluator based on the correlation proposed by Hall (1999); moreover, a search criterion that followed a hill climbing algorithm with

backtracking was utilized. Such combination produced a significant increase in accuracy.

Specific parameters were established to select attributes in WEKA, as follows: the evaluator was CfsSubSetEval (numThreads 1, poolSize 1) and the search strategy was BestFirst (direction: Forward, searchTermination 5).

Additionally, other search strategies, such as ExhaustiveSearch, were tried, but the verification of all the possibilities of a large set of attributes was highly demanding in terms of time. In like manner, genetic search was resorted to; although it performed faster, results were not good, even with the increase in the number of generations, population size and the adjustment in recombination/mutation probability.

Eventually, BestFirst was more efficiently applied to the representation of the documents used in this work. A more detailed study of all the approaches to select attributions may produce better results.

At once, it was found that the binarization of the vectors of characteristics turned into a more precise model to detect deceptive text. Table IV.5 shows the results of deception detection with characteristics LDA+TDM and the effect of applying binarization and attribute selection; binarization was implemented before attribute selection. Since the use of converting binary values as a set with the selection of the attributions produced the best results in all the cases, this combination was used in all coming experiments. The outputs (precision, recall and F-measure) of binarization with attribute selection (Binary with AtSe) indicated in table IV.5 are shown in section 5.

Following, the conversion of all the characteristics into binary values is detailed.

- LIWC generated 64-characteristic vectors. The means to obtain each vector were the following: given a document and the 64 categories, if a word in a current category was found in the document, then such characteristic had a value of one, on the contrary, it was zero.

- The means to generate vectors of characteristics using ns-grams was the following: in the first place, a list of non-repeated ns-grams obtained from all the documents of the dataset to analyze was created. Secondly, given a document and the ns-gram list, if the present ns-gram was found in the document, then the value of the characteristic was established at 1; on the contrary, it was established at 0.

- As previously mentioned, LDA shows, as a result, vectors of characteristics with actual values (probability to belong to each topic). Thereby, we converted the characteristics' values into binary values. To do so, a threshold was calculated dividing the addition of all the probabilities to belong by the number of established topics. Each probability equal or greater than the threshold became 1; on the contrary, zero.

Table IV.5: Comparison of accuracy (%) regarding binary values and attribute selection (AtSe)

Corpus	Binary+AtSe	Non-binary+AtSe	Binary w/no AtSe
Abortion	87.5	75.2	72.2
Best friend	87.0	82.0	76.8
Capital punishment	80.0	69.3	61.5
DeRev	94.9	74.9	88.8
OpSpam	90.9	88.5	87.2

4.4. Identification of the optimal number of topics

LDA needs the specification of topics to be generated; any change to this parameter may change the accuracy of classification. Owing to this, it is necessary to find an appropriate value.

Table IV.6: F-measure using different values for number of topics

Topics	*Abortion*	*Best friend*	*Capital punishment*	DeRev	OpSpam
100	69.46	78.01	57.75	81.25	87.36
200	75.52	82.61	65.18	86.47	87.93
300	81.17	85.38	71.71	90.20	89.93
400	84.50	87.05	77.22	92.35	90.06
500	**86.70**	**87.58**	**80.41**	**94.53**	**90.10**
600	87.40	87.00	81.18	95.54	91.34
700	87.52	86.29	81.86	95.74	89.53
800	87.76	85.03	82.05	95.68	90.15
900	87.38	84.33	81.82	95.97	90.68
1 000	86.92	83.29	81.72	96.06	90.25

To find the optimal number of topics that allow an optimal classification, several experiments were carried out; these experiments' results are displayed in table IV.6, the number of topics is compared with the F-measure obtained. As well, it is noticed that as the number

of topics increases, it is possible to reach an optimal point for which the increase in the number of them does not imply a diminution in F-measure (this is to say, 500 topics).

Although it is possible to slightly increase the accuracy of some datasets by means of adjusting the number of topics, the goal was to find a general model with characteristics that deliver good performance with various sorts of deception. This way, a set of common parameters that work for all datasets was sought.

Due to the foregoing, all the experiments used 500 topics, i.e., each document processed by LDA generates a 500-characteristic vector and each is represented by a probability of belonging to each topic (section 4.1.4, figure IV.5).

Chapter 5

Assessment of the approach proposed

5.1. Detection of deception

This section presents in detail the results of deception detection for various methods of generation of characteristics. Tables for each classified corpus show the following values: accuracy, precision (P), recall (R) and F-measure (F).

Although accuracy is a measure used in many researches on deception detection and serves as a point of comparison with other results, precision, recall and F-measure were as well included, for this enables a deeper analysis of results. This way, precision shows the percentage of selected texts that are correct; while recall, the percentage of correct texts that are selected. Finally, F-measure is the combined measure to assess P/R compensation.

The aforementioned measures were obtained for the following methods: Latent Dirichlet Allocation (LDA), a word space model (TDM, Term-Document Matrix), syntactic n-grams (ns-grams) and Linguistic Inquiry and Word Count (LIWC); as well as combinations of them.

To test the efficiency of each method, three datasets on various topics were processed: OpSpam, opinions on hotels; DeRev, opinions on books purchased on Amazon; and, the dataset of controversial topics, composed of opinions on three topics (*abortion*, *capital punishment* and *best friend*).

On the basis of the available datasets, it was researched which method was best on a single domain (sec. 5.1.2), where both the training and testing sets dealt with the same topic in a mixed environment domain (sec. 5.1.3); where both the training and testing sets are a mix of topics; and, in a crossed-domain environment (sec. 5.1.4), in which the training and testing sets are composed of different topics (with no intersection between testing and training).

Finally, it was demonstrated that the approach proposed for automatic deception detection may be used in authorship identification (sec. 5.3), obtaining competitive results. Moreover, it was verified that the method to generate characteristics is independent from domain and language, since in this second task texts in various languages from different domains and genres are analyzed.

5.1.1. Validation measures

When proposing a method to automatically detect deceptive text, a measure to validate the results is also needed; that is to say, a measure that delivers a numerical value that is correlated to the efficiency of the method proposed.

A measure broadly accepted owing to its robustness in information retrieval is F-measure; this measure indicates the efficiency of classification. Likewise, it is also common to use the area under the

curve, which indicates to what extent the classifier has the capacity to discriminate different-class elements.

Both provide an objective way to prove how well the method proposed detects deceptive texts in a collection.

5.1.1.1. F-measure

On the one side, it is a harmonic measure that comprises two proportions: recall and precision. This measure reaches its best value at 1, where perfect precision and recall are attained. Likewise, as noticed in figure V.1, this measure takes various aspects of a classification into account.

$$Precision = \frac{TP}{TP + FP} \tag{5.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.2}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.3}$$

Furthermore, precision focuses on the selected documents (eq. 5.1); while recall takes all the relevant documents into consideration (eq. 5.2).

In this way, F-measure considers well-selected documents, but also the documents which had to be selected and were discriminated (eq. 5.3).

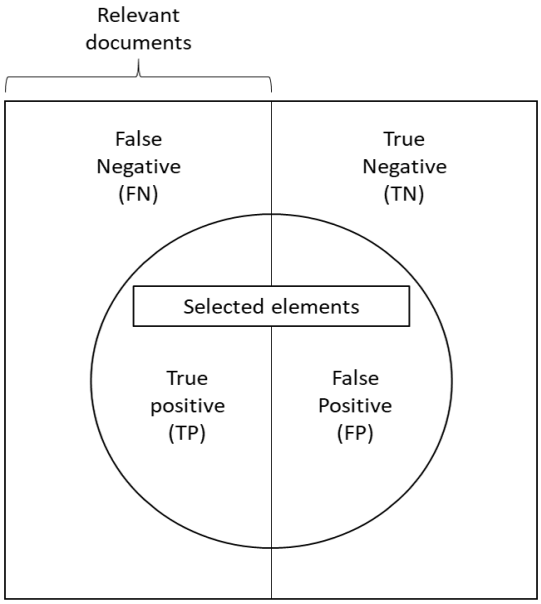


Figure V.1: Precision and recall

5.1.1.2. Area under the ROC curve

ROC (Receiver Operating Characteristic) curve indicates how good a binary classifier is at distinguishing elements of two classes; for instance, if a text is deceptive or truthful.

This measure involves the aspects of a classification task, shown in figure V.1, and is based on two measures: true positive rate (TPR) and false positive rate (FPR), obtained through equations 5.4 and 5.5, respectively. TPR and FPR are used in the generation of the ROC curve to finally measure the bidimensional area under the curve (AUC). AUC result is the indicator of the classifier performance.

$$TPR = \frac{TP}{TP + FN} \tag{5.4}$$

$$FPR = \frac{FP}{FP + TN} \tag{5.5}$$

5.1.2. Classification of a specific domain

The goal of combining various sources of generations of characteristics (section 4.1) is to find which combination favors accuracy and F-measure in the classification of deceptive and truthful texts best.

Words contain much information on their own, this can be verified by the large number of research works (section 3) which, with a view to obtaining a better performance, use unigrams as a base to add new characteristics.

Similarly, in this study LDA was utilized to complement the characteristics of TDM.

This chapter's results show that the characteristics generated by means of TDM in combination with those of LDA obtain better results than other methods (such as LIWC, ns-grams) in most of the cases.

In the classification of OpSpam, shown in table V.1, it is noticed that the combination of LDA and TDM shows an increase in precision regarding the same techniques assessed separately. This is due to the fact that the characteristics generated using LDA complement more favorably those generated by TDM.

There exist cases in which there are many characteristics that represent truthful text, whereas only some characteristics account for deceptive text or vice versa. LDA may help minimize this problem

increasing the number of relevant characteristics that lead to a more efficient text classification.

To study this effect, the number of attributes obtained was numbered after implementing the technique to select attributes that are closely linked to deceptive or truthful text; for example, in 60 relevant characteristics, there is a specific number of characteristics closely related to deception, while other characteristics to truthful texts (see the two first TDM columns in table V.2).

Moreover, the number of relevant characteristics may increase combining methods of generation of characteristics; table V.2 shows the relevant characteristics obtained for each class using only TDM in comparison with the number of characteristics obtained using the combination of LDA and TDM. It is noticed that in all cases there is an increase in deceptive text's characteristics.

The vectors of characteristics that were generated by the combination of LIWC and TDM characteristics showed similar results, though with an inferior performance in comparison with LDA+TDM combination. However, the main disadvantage, regarding LDA, is that LIWC is labeled by humans; this means that every time a different language is analyzed, it is necessary to obtain the tool for that specific language. On the other side, LDA does not require any language-dependent information to analyze a dataset.

Another approach was the use of ns-grams. As with LIWC, previous language-dependent information is necessary, since parse trees are obtained on the basis of the probability of occurrence in a previously tagged corpus.

Syntactic bigrams showed acceptable precision; however, the combination of their characteristics may worsen the results. For example, in the datasets in tables V.3 and V.5, the combination of characteristics between ns-grams and TDM decreased the accuracy of classification. Therefore, this specific combination of characteristics was disadvantageous.

It is important to point out that the method proposed (LDA+TDM) remains in competitive accuracy to detect deception in the various datasets classified, unlike some other NLP techniques such as combination between syntactic relations and unigrams used by Feng *et al.* (2012).

The result of the authors is good at assessing OpSpam, which is shown in table V.1. However, this performance was not consistent when the authors classified data on *abortion*, *best friend* and *capital punishment*; as displayed in tables tables V.4-V.6, respectively.

On the basis of the results obtained, it was found that the best combination of characteristics was LDA and term-document matrix (LDA+TDM). This combination shows an accuracy of 90,9 % for OpSpam and 94,9% for DeRev. Accuracy values of 87.5 %, 87.0% and 80.0% for *abortion*, *capital punishment* and *best friend*, respectively, were obtained.

Figure V.2 shows the learning process of each dataset. For each percentage of randomly sampled data, from 10 to 100%; out of these range of percentages, 80% of them was used as a training set, while the other 20%, as a testing set. For example, if 60% of the data is used, i.e., 600 out of one thousand records; then, 480 records will be used for training and 120 for testing. This graph suggests that for some corpora, the use of more data may lead to a small increment in

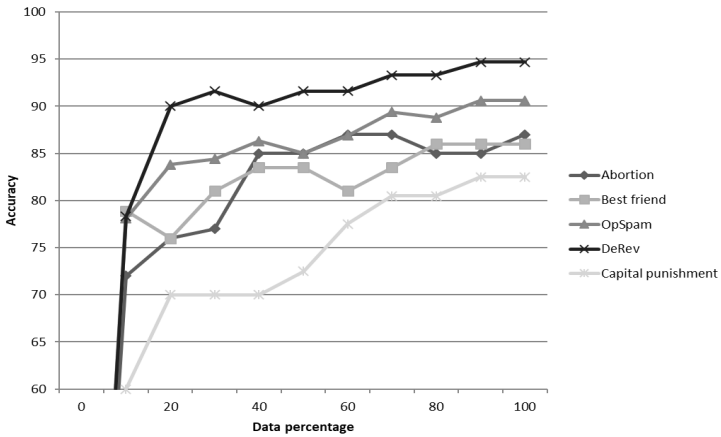


Figure V.2: Learning curve for the various corpora

accuracy, while for others the increase between 90% or 100% of all the data is not considerable.

5.1.3. Classification of mixed domains

The main objective of the experiments shown below was to research the performance in classification when datasets for deception detection from various domains are combined. With this, the training dataset would contain topics also in the testing dataset (but not single-domain texts).

In this case, once again a combination of characteristics of LDA and TDM produced the best result (table V.7). In table V.8, it is seen that SVM had a better performance than the NB classifier.

Table V.1: OpSpam classification by means of various sources of generations of characteristics

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	72.0	72.8	70.3	71.5	71.3	73.8	72.5
TDM	87.9	87.0	89.0	88.0	88.7	86.8	87.7
ns-grams (brigrams)	68.6	72.8	59.5	65.5	65.8	77.8	71.2
ns-grams + TDM	86.0	88.7	82.5	85.5	83.6	89.5	86.5
LIWC	65.0	65.7	62.8	64.2	64.4	67.3	65.8
LIWC + TDM	87.9	89.4	86.0	87.6	86.5	89.8	88.1
LDA + TDM	90.9	94.8	86.5	90.5	87.6	95.3	91.3
LDA + LIWC	72.3	73.9	68.8	71.2	70.8	75.8	73.2
LDA + ns-grams	74.9	72.6	80.0	76.1	77.7	69.8	73.5
LDA + TDM + LIWC	87.1	87.2	87.0	87.1	87.0	87.3	87.1
LDA + LIWC + ns-grams	79.7	77.8	83.3	80.4	82.0	76.3	79.0
LDA + TDM + ns-grams	88.9	88.6	89.3	88.9	89.2	88.5	88.8
LDA + TDM + LIWC + ns-grams	88.1	88.2	88.0	88.1	88.0	88.3	88.1
Ott *et al.* (2011) (LIWC + brigrams)	89.8	89.8	89.8	89.8	89.8	89.8	89.8
Feng *et al.* (2012) (syntactic rel. + unigrams)	91.2	-	-	-	-	-	-
Fusilier *et al.* (2015)	90.2	-	-	-	-	-	-

Table V.2: Number of relevant characteristics obtained from the combination of LDA and TDM

	TDM		TDM+LDA	
Corpus	Truthful	Deceptive	Truthful	Deceptive
Abortion	43	12	45	25
Best friend	24	17	30	20
Capital punishment	32	16	39	29
DeRev	27	52	37	57
OpSpam	31	22	31	24

Table V.3: DeRev classification by means of various sources of generation of characteristics

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	80.5	79.0	83.1	81.0	82.1	78.0	80.0
TDM	91.9	90.9	93.2	92.1	93.0	90.7	91.8
ns-grams (brigrams)	75.4	80.0	67.8	73.4	72.1	83.1	77.2
ns-grams+TDM	91.1	89.4	93.2	91.3	92.9	89.0	90.9
LIWC	69.9	77.0	56.8	65.4	65.8	83.1	73.4
LIWC + TDM	91.5	92.2	90.7	91.5	90.8	92.4	91.6
LDA+TDM	94.9	94.2	95.8	95.0	95.7	94.1	94.9
LDA+LIWC	73.3	81.6	60.2	69.3	68.5	86.4	76.4
LDA+ns-grams	78.4	83.2	71.2	76.7	74.8	85.6	79.8
LDA+TDM+LIWC	90.7	98.0	83.1	89.9	85.3	98.3	91.3
LDA+LIWC+ns-grams	79.2	86.3	69.5	77.0	74.5	89.0	81.1
LDA+TDM+ns-grams	91.5	90.2	93.2	91.7	93.0	89.8	91.4
LDA+TDM+LIWC +ns-grams	83.0	84.8	80.5	82.6	81.5	85.6	83.5
Fornaciari and Poesio (2014)	76.27	-	-	-	-	-	-

5.1.4. Crossed domain classification

Unlike the classification that combines all the domains, for this experiment every dataset was selected for testing and the rest were combined as a training dataset. This way, the training dataset's domain was not included in training.

Table V.9 shows the results for interdomain classification. In these experiments, unlike those presented in sections 5.1.2 and 5.1.3, the combination of LDA and TDM characteristics consistently did not produce the best accuracy.

In most of the cases (3 in 5), SVM surpassed NB; however, with a relatively simple configuration of a term-document matrix, NB is capable

Table V.4: Classification of *abortion* using various sources of generation of characteristics

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	79.5	76.6	85.0	80.6	83.1	74.0	78.3
TDM	82.5	82.8	82.0	82.4	82.2	83.0	82.6
ns-grams (brigrams)	65.8	73.2	52.0	60.8	62.8	81.0	70.7
ns-grams+TDM	83.0	83.0	83.0	83.0	83.0	83.0	83.0
LIWC	72.5	72.7	72.0	72.4	72.3	73.0	72.6
LIWC + TDM	83.5	81.3	87.0	84.1	86.0	80.0	82.9
LDA+TDM	87.5	87.9	87.0	87.4	87.1	88.0	87.6
LDA+LIWC	80.0	76.8	86.0	81.1	84.1	74.0	78.7
LDA+ns-grams	73.0	78.8	63.0	70.0	69.2	83.0	75.5
LDA+TDM+LIWC	76.5	76.2	77.0	76.6	76.8	76.0	76.4
LDA+LIWC+ns-grams	77.0	76.5	78.0	77.2	77.6	76.0	76.8
LDA+TDM+ns-grams	86.5	90.1	82.0	85.9	83.5	91.0	87.1
LDA+TDM+LIWC +ns-grams	82.5	77.8	91.0	83.9	89.2	74.0	80.9
Pérez Rosas and Mihalcea (2014a)	80.3	-	-	-	-	-	-
Feng *et al.* (2012)(syntactic rel. + unigrams)	77.0	-	-	-	-	-	-

to improve deception detection with the characteristics learned from other datasets.

5.1.5. Prevailing words and topics

There have been efforts to improve detecting deception in a text by means of various techniques. Such task is not easy since the characteristics that define a deceptive text may vary between datasets.

Some words that represent false text in a dataset may represent truthful text in another; this indicates that a word does not strictly represent one class or another in all cases. For example, in table V.12

Table V.5: *Best friend* classification by using various sources of generation of characteristics

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	82.0	82.7	81.0	81.8	81.4	83.0	82.2
TDM	85.5	84.5	87.0	85.7	86.6	84.0	85.3
ns-grams (brigrams)	73.0	81.1	60.0	69.0	68.3	86.0	76.1
ns-grams+TDM	84.5	84.8	84.0	84.4	84.2	85.0	84.6
LIWC	75.5	74.3	78.0	76.1	76.8	73.0	74.9
LIWC + TDM	83.5	82.5	85.0	83.7	84.5	82.0	83.2
LDA+TDM	87.0	85.6	89.0	87.3	88.5	85.0	86.7
LDA+LIWC	78.0	73.7	87.0	79.8	84.1	69.0	75.8
LDA+ns-grams	85.0	88.9	80.0	84.2	81.8	90.0	85.7
LDA+TDM+LIWC	77.0	72.5	87.0	79.1	83.8	67.0	74.4
LDA+LIWC+ns-grams	79.5	78.6	81.0	79.8	80.4	78.0	79.2
LDA+TDM+ns-grams	85.0	92.7	76.0	83.5	79.7	94.0	86.2
LDA+TDM+LIWC +ns-grams	86.5	94.0	78.0	85.2	81.2	95.0	87.6
Pérez Rosas and Mihalcea (2014a)	75.9	-	-	-	-	-	-
Feng *et al.* (2012) (syntactic rel. + unigrams)	85.0	-	-	-	-	-	-

it is observed that personal pronoun *I* and possessive adjective *my* are characteristics of truthful text in the dataset *abortion*; conversely, as seen in table V.10, these two words work as characteristics of deceptive text in OpSpam.

The fact that certain words are contradictory in different datasets does not indicate that automatic methods to recognize deceptive text are dysfunctional, but such words may represent a specific dataset, but will not appear in a more general set, for example when joining OpSpam with the *abortion* set.

Table V.6: *Capital punishment* classification using various sources of generation of characteristics

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	77.0	76.5	78.0	77.2	77.6	76.0	76.8
TDM	78.5	77.7	80.0	78.8	79.4	77.0	78.2
ns-grams (brigrams)	61.5	66.2	47.0	55.0	58.9	76.0	66.4
ns-grams+TDM	78.5	78.8	78.0	78.4	78.2	79.0	78.6
LIWC	65.5	64.5	69.0	66.7	66.7	62.0	64.2
LIWC + TDM	78.0	78.6	77.0	77.8	77.5	79.0	78.2
LDA+TDM	80.0	78.8	82.0	80.4	81.3	78.0	79.6
LDA+LIWC	66.0	79.6	43.0	55.8	61.0	89.0	72.4
LDA+ns-grams	70.0	72.7	64.0	68.1	67.9	76.0	71.7
LDA+TDM+LIWC	61.0	60.4	64.0	62.1	61.7	58.0	59.8
LDA+LIWC+ns-grams	75.0	78.4	69.0	73.4	72.3	81.0	76.4
LDA+TDM+ns-grams	74.0	91.4	53.0	67.1	66.9	95.0	78.5
LDA+TDM+LIWC +ns-grams	79.0	73.4	91.0	81.3	88.2	67.0	76.1
Pérez Rosas and Mihalcea (2014a)	77.2	-	-	-	-	-	-
Feng *et al.* (2012) (syntactic rel. + unigrams)	71.5	-	-	-	-	-	-

Table V.7: Accuracy, precision (P), recall (R) and F-measure (F) obtained with the corpora mixed by means of SVM

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	65.8	67.6	60.9	64.1	64.4	70.8	67.4
TDM	73.5	73.6	73.5	73.5	73.5	73.6	73.5
LDA + TDM	76.3	76.8	75.6	76.2	75.9	77.1	76.5
LIWC	59.8	60.1	58.4	59.2	59.5	61.1	60.3
LIWC + TDM	73.7	71.9	77.8	74.7	75.8	69.7	72.6
LDA + LIWC	69.7	69.3	70.8	70.1	70.2	68.7	69.4
LIWC + TDM + LDA	72.6	72.3	73.5	72.9	73.0	71.9	72.5

Table V.8: Accuracy, precision (P), recall (R) and F-measure (F) obtained with the corpora mixed by means of NB

		Truthful			Deceptive		
Method	Accuracy	P	R	F	P	R	F
LDA	61.1	61.5	59.3	60.4	60.7	62.8	61.7
TDM	74.3	73.9	75.3	74.6	74.8	73.3	74.1
LDA + TDM	73.5	71.9	77.0	74.4	75.3	69.9	72.5
LIWC	56.4	55.6	64.7	59.8	57.7	48.3	52.6
LIWC + TDM	73.3	73.3	73.3	73.3	73.3	73.3	73.3
LDA + LIWC	64.8	65.1	64.2	64.6	64.7	65.5	65.1
LIWC + TDM + LDA	73.3	76.0	68.3	71.9	71.2	78.4	74.6

Table V.9: F-measure from the cross-domain classification. Results for NB and SVM are displayed

		DeRev	OpSpam	*Abortion*	*Best friend*	*Capital punishment*	Average
LDA	SVM	52.1	48.8	57.5	50.0	53.5	50.43
	NB	43.2	49.8	56.0	51.5	56.0	46.50
TDM	SVM	53.3	52.8	55.5	55.5	58.5	53.05
	NB	50.8	53.8	58.5	56.0	48.5	52.30
LIWC	SVM	51.3	49.2	53.1	54.3	52.8	50.25
	NB	50.9	51.1	52.6	56.8	51.4	51.00
LIWC + TDM	SVM	54.6	53.8	54.5	55.0	56.0	54.20
	NB	47.8	52.5	55.5	59.0	51.0	50.15
LDA + TDM	SVM	59.3	50.6	57.5	64.0	55.0	54.95
	NB	58.8	52.3	55.0	59.5	52.5	55.55
LDA + LIWC	SVM	56.3	46.3	54.5	55.5	52.0	51.30
	NB	45.7	48.1	46.5	51.5	49.5	46.90
LDA+LIWC +TDM	SVM	52.1	52.6	57.0	57.0	54.0	52.35
	NB	52.9	53.0	58.0	62.5	53.0	52.95
Best:		SVM	NB/SVM	NB	SVM	SVM	NB

It was verified that the combination of LDA and TDM characteristics had good accuracy in the three studied datasets. This implies that models generated by this combination of characteristics are more reliable to detect deceptive text, since they have been tried in cases of deception and have remained stable as regards the accomplished accuracy.

Moreover, LDA+TDM may be applied to other languages as they do not require any corpus to produce characteristics such as LIWC or syntactic n-grams.

On the one side, the use of LDA was to a good extent a more efficient alternative method than LIWC. This is because LDA generates topics and considers all the words in the documents in such manner that no word will escape from the statistical process of generation of groups of words. On the other side, LIWC contains a group of preestablished words, of which some are probably not included in the documents to be processed. Thereby, the generation of a model that represents deceptive text may be more successful than using LDA.

In order to show how the combination of LDA and TDM is favorable, table V.12 includes some examples of the most relevant words and topics in both deceptive and truthful texts in *abortion*. There are two distinguishable aspects in this instance. On one side, topics are dominant characteristics in detecting deceptive texts; this is say, it comprises more topics than single words. While, on the other, words are dominant characteristics in truthful text.

It was noticed, in the analyzed datasets, that if a class lacks relevant characteristics that further stress the difference between classes, there will be topics that increase such characteristics and improve accuracy in classification. Tables V.11, V.13 and V.14 contain highly

Table V.10: The 10 most relevant topics and words, representing deceptive and truthful texts in OpSpam

Deceptive text	Truthful text
my	*location*
I	*floor*
luxury	*block*
relax	*bathroom*
spa	*street*
visit	*large*
vacation	*small*
anyone	2
amaze	*priceline*
chicago	*upgrade*

discriminating topics and words between deceptive and truthful texts. Topics can be identified as a set of independent words, which must not be mistaken as a sentence.

5.1.6. Comparison of results and statistical significance

Comparing results is important since it demonstrates the effectiveness of the approach proposed, in addition to show the advantages and disadvantages regarding other approaches studied.

Table V.15 shows a comparison between the method proposed and the results of various researchers, who used the same datasets from various perspectives. It is noticed that the approach proposed in this study had a better performance in the classification of most datasets, except for OpSpam.

However, a better performance or accuracy do not mean there exists statistical significance; that is to say, if the increase in precision is

Table V.11: The 10 most relevant topics and words, representing deceptive and truthful texts in *best friends*

Deceptive text	Truthful text
he, is, guy, really, up, attention, overall, unintentionally, quite	*we*
this	*each*
his	*fried*
guy	*have*
lie	*same*
nice	*year*
put	*my*
trustworthy	*we, each, other, have, interests, but, over, one, stuff, others*
wonderful	*live*
how	*good*

sufficiently significant to conclude that the approach is better or simply obtains results equivalent to those from other methods.

For that reason, table V.16 shows the statistical significance between this research's results and those of other researches. With a view to comparing, an (α) significance level of 0.05 (5%) is obtained, which means that statistical significance is reached if p value is smaller than α. With this significance level, the performance of some of our results would not have statistical significance in comparison with other existing methods, making them virtually equivalents.

Table V.12: The 10 most relevant topics and words, representing deceptive and truthful texts in *abortion*

Deceptive text	Truthful text
god	*I*
to, of, end, moment, cannot, give, consequences, society, perfect, return, couples, chance, that	*but*
is, not, this, womb, human, piece, so, there, inches, carry, conception, trash	*believe*
chance	*choice*
action	*or*
necessary	*me*
we, and, can, this, society, an, may, base, serve, contribute, nerves, started, effect, removed, his, always, subject	*for*
morally	*I, feel, life, me, know, even, people, bit, ends, experience, especially, unborn, control, sex, mother*
they, actions, believe, effectively, seriously, eyes, outcome, ask, mirror, ready	*my*
who, way, these, perform, themselves, where, example, available, always, out	*without*

5.2. Impact of polarity on deceptive texts

By and large, deceptive opinions may be positive or negative. The former aim at persuading buyers to choose a product; while the latter try to smear a product's reputation.

Table V.13: The 10 most relevant topic and words, representing decptive and truthful texts in *capital punishment*

Deceptive text	Truthful text
instead-of	*I*
they, wouldn't, repetitive, capital, offenders, horrid, are, have, and, the	*believe*
him	*I, the, death, believe, penalty, that, am, waste, sufficient, wrongly, silly, because, be*
would, that, cases, also, upon, rape, used, get, someone, this, give, which, criminals	*put*
be, an, on, should, arm, everybody, individual, he, alive, year, forgive, concern, love, understand, please, had	*death*
it, people, justice, prison, first, severe, shot, trials, major, dictate, killed, trial	*for*
lesson	*opinion*
possibility	*sure*
the, because, humanely, promotes, per, population, future, offenders, equivalent, didn't, murderer, shows	*of*
people, as, murder, that, humankind, off, decide, alive, instead, undone, person	*current*

Thus far, there are some studies that have attempted to discover universal signals of deception (DePaulo *et al.*, 2003); though there is no universal set of characteristics able to identify deception in text.

Table V.14: The 10 most relevant topics and words, representing deceptive and truthful text in DeRev

Deceptive text	Truthful text
author	*the, was, i, they, were, some, which, had, not, during, these, even, when, if, because, character, been, most, several*
your	*i, the, it, book, was, read, but, reading, first, all, my, had, this, that, one, so, characters, love*
uno/un	*history, american, native, our, americans, country, heart, indians, west, day, newspaper, the, tone, brown, better, known, fully, resist, stolen*
thriller	*classic*
the, and, of, a, to, is, in, that, this, for, book, as, it, on, with, an, are, but	*edition*
and	*I*
intrigue	*the, movie, book, than, seen, made, much, movies, version, like, cover, film, better, more, do, it, least, half, then*
guide	*version*
drug	*history*
issue	*again*

This suggests that signals of deception may vary depending on the nature of the texts; owing to this, experiments in this section will be run in order to account for characteristics, under the supposition that

Table V.15: Comparison of our results with other studies on the same corpora

Corpus	Studies	Accuracy
OpSpam	This study (LDA+TDM)	90.90%
	Ott *et al.* (2011) (LIWC + brigrams)	89.80%
	Feng *et al.* (2012) (syntactic rel.+ unigrams)	91.20%
	Fusilier *et al.* (2015)	90.20%
DeRev	This study (LDA+TDM)	94.90%
	Fornaciari and Poesio (2014)	76.27%
Abortion	This study (LDA+TDM)	87.50%
	Pérez Rosas and Mihalcea (2014a)	80.30%
Best friend	Feng *et al.* (2012) (syntactic rel. + unigrams)	77.00%
	This study (LDA+TDM)	87.00%
	Pérez Rosas and Mihalcea (2014a)	75.90%
	Feng *et al.* (2012) (syntactic rel. + unigrams)	85.00%
Capital punishment	This study (LDA+TDM)	80.00%
	Pérez Rosas and Mihalcea (2014a)	77.20%
	Feng *et al.* (2012) (syntactic rel. + unigrams)	71.50%

taking the texts' polarity into account may increase accuracy in detecting deceptive text. Deceptive opinions have been analyzed in a number or research works, but commonly the authors display general classifications; albeit, many characteristics may be discarded since deceptive opinions can be positive or negative.

That is to say, negative opinions may include particular characteristics that can difference only that category of opinions; therefore, it is possible that the texts' polarity directly affects the characteristics that help identify truth or deception.

Over the analysis of various datasets, the impact of polarity on the classification of deception in different domains was as well analyzed. The same available datasets were used for carrying out relevant tests and a number of experiments were run in order to verify to what extent certain characteristics are specific to each polarity (positive and negative).

Table V.16: Statistical significance

Corpus	# docs.	This study	Other studies	p-value	Stat. signific.
OpSpam	800	90.9%	Ott *et al.* (2011) (89.9%)	0.248	No
			Feng *et al.* (2012) (91.2%)	0.416	No
			Fusilier *et al.* (2015) (90.2%)	0.316	No
DeRev	236	94.9%	Fornaciari and Poesio (2014) (76.3%)	0	Sí
Abortion	200	87.5%	Pérez Rosas and Mihalcea (2014a) (80.3%)	0.025	Sí
			Feng *et al.* (2012) (77.0%)	0.003	Sí
Best friend	200	87.0%	Pérez Rosas and Mihalcea (2014a) (75.9%)	0.002	Sí
			Feng *et al.* (2012) (85.0%)	0.283	No
Capital punishment	200	80.0%	Pérez Rosas and Mihalcea (2014a) (77.2%)	0.248	No
			Feng *et al.* (2012) (71.5%)	0.023	Sí

Likewise, a polarity classifier was implemented as a preprocessing stage for a specific classification. The results disclose there are specific signals of deception, which may vary depending on the texts' polarity; additionally, an improvement in most of the classified datasets is noticed when a polarity classifier is used.

In the first studies, the combination of LDA+TDM produced the best results in detecting deception in general. However, since false opinions may be positive and negative, it may have been possible that the classification heavily depended on the opinions' polarity.

For example, a corpus that comprises 100 deceptive and 100 truthful opinions; for their part, the former may be largely positive, while the latter mostly negative. Accordingly, it was supposed that the classifier may tend to identify characteristics that correspond to the texts' polarity instead of those corresponding to deception in the texts.

To discard the supposition above, a simple experiment was carried out. Assuming that the precision of classification should decrease if the presence of polarity words were linked to opinion classification, positive and negative words were excluded using a polarity lexicon (Hu and Liu, 2004).

Such lexicon was created by means of the processing a set of opinions on certain products and it is in constant growth. At present, it holds 2 006 positive and 4 783 negative words selected out of subjective opinions. The authors selected the opinions based on the fact that the presence of adjectives may define the expression of an opinion; moreover, they restrict the selection to the opinions produced from sentences related to the products' quality.

As noticed in table V.17, F-measure obtained after excluding polarity words barely decreased regarding the F-measure obtained with the original corpora. Words on their own may be only indicators of the texts' polarity, but not a defining factor to identify deception. Therefore, these results show that classification does not depend on positive or negative words as relevant characteristics.

On the one side, the fact that positive and negative words do not affect the results indicates that providing vectors with polarity is not the best way to improve deception detection.

On the other, based on the fact that there is some evidence that the characteristics of deception may vary depending on the texts' nature (Hernández Castañeda and Calvo *et al.*, 2017), it was expected to improve deception detection making a change in the classification of polarity before capturing the characteristics.

Table V.17: Results of the classification excluding polarity words

Corpus	P	R	F-measure	F-measure, unchanged corpus
Abortion	0.871	0.870	0.870	0.875
Best friend	0.87	0.870	0.870	0.870
Capital punishment	0.819	0.815	0.814	0.800
OpSpam	0.865	0.865	0.865	0.867
DeRev	0.945	0.945	0.945	0.949

Table V.18: Classification of OpSpam, tagged by the authors (OAP) and that of the present study (PTW), based on the text's polarity

Polarity	# docs.	P	R	F
POSITIVE-OAP	800	0.912	0.909	0.910
NEGATIVE-OAP	800	0.844	0.844	0.844
POSITIVE-PTW	620	0.828	0.826	0.826
NEGATIVE-PTW	980	0.843	0.843	0.843
POS+NEG	1600	0.867	0.867	0.867

Table V.19: Classification of the datasets adding the polarity feature

Corpus	P	R	F
Abortion	0.871	0.870	0.870
Best friend	0.865	0.865	0.865
Capital punishment	0.815	0.810	0.809
DeRev	0.928	0.928	0.928
OpSpam	0.865	0.865	0.865

Table V.20: Comparison of F-measure before (F-Ant) and after (Avg. F) polarity classification. Precision (P), Recall (R), F-measure (F) and the number of opinions by class (# docs.) are displayed as well

Corpus	Positive				Negative					
	docs.	P	R	F	docs.	P	R	F	Avg-F	F-Ant
Abortion	49	0.931	0.918	0.919	151	0.895	0.894	0.894	0.906	0.875
Best friend	121	0.868	0.868	0.868	79	0.863	0.861	0.861	0.864	0.870
Capital punishment	87	0.885	0.885	0.885	113	0.868	0.867	0.867	0.867	0.800
DeRev	158	0.930	0.930	0.930	78	0.987	0.987	0.987	0.958	0.949
OpSpam	620	0.826	0.826	0.826	980	0.843	0.843	0.843	0.835	0.867

5.2.1. Polarity classification to improve deception detection

The results in section 5.2 may suggest that the classifier does not consider characteristics related to polarity. However, as previously stated, some evidence indicates that polarity may be useful to identify deception in the text. Therefore, despite polarity words do not directly influence the classification of deceptive text, such words may indirectly help as they previously classify positive and negative opinions (as a preprocessing stage) and then classify opinions as truthful or deceptive.

At this point, the characteristics generated would be specific for each class of polarity. Owing to this, a partial classification was carried out instead of a general one. This is to say, the opinions are first divided according to their polarity to improve the classification results. As a first assessment of this strategy, OpSpam was used since it had been already divided into positive and negative opinions (Ott *et al.*, 2013).

The classification based on OpSpam's polarity, shown in table V.18, consisted in classifying positive, negative opinions and the combination of them. At this point, there are two methods to divide the corpus, OpSpam, by polarity: the authors' classification of OpSpam (OAP) (Ott *et al.*, 2013) and this work's classification (PTW). The former, as it was classified by the authors, is considered a golden standard. OAP records show the results of the classification bearing in mind the division of OpSpam's authors, while PTW records show the results with the corpus divided by a polarity classifier developed in the present study.

With a view to building a supervised classifier of positive and negative opinions, we followed Pang *et al.* (2002); this way, unigrams were obtained from a corpus of film reviews. Then, they were rearranged by frequency to produce a dictionary with 16 165 elements. For the training set, each film review was read to make a vector with the words' frequencies; that is to say, if the word exists, such characteristic increases its counting. Finally, a Bayesian classifier was used to process vectors. This method based on machine learning only includes lexical information; however, it has produced acceptable results. Exploring other approaches to classify polarity may improve efficiency in detecting deception.

The results evince that the polarity classifier could not have produced an optimal classification, as the average of F-measure for PTW is smaller than for OAP. Thereby, if this behavior remains, the classification results for the rest of the corpora may be improved using a more robust polarity classifier.

This classifier was applied to the remaining datasets, in such manner that each of them is now divided into positive and negative opinions (and each class has its truthful and deceptive opinions, respectively). Table V.20 shows the number of positive and negative opinions for each corpus. By means of this experiment, the method proposed is put to the test and it is verified there are certain improvements in classification.

To compare the effectiveness of a previous polarity classification and simply adding polarity characteristics to the vectors, a second experiment was carried out; in which information on polarity was added as a characteristic. This was included in the patterns before the classification of the corpora; the difference is that in this case, the entire corpus is considered, and another characteristic is added: if the opinion is positive or negative.

To define polarity, the classifier generates a real value between 0 and 1; this way, the value of the characteristic will be 1 if the real value is above 0,5; on the contrary, it will be 0.

As noticed in table V.19, on the one side, F-measure barely changes in the case of controversial topics (*abortion*, *best friend* and *capital punishment*); on the other, the classification of DeRev and OpSpam showed that F-measure decreased to a larger extent. These results indicate that including polarity information in the patterns

(as an additional characteristic) is not useful and even detrimental to deception detection.

All of the above demonstrates that obtaining specific characteristics by dividing the texts into positive and negative before looking for signals of deception (approach proposed in the present work, see table V.20) produced better results than simply adding polarity information as an additional characteristic (table V.19).

5.3. Solving a different task

Since the proposed approach is independent from domain and language, it is possible to use it in other tasks in which machine learning algorithms may produce models based on the chosen dataset.

This section deals with how LDA can be applied to various classification tasks.

5.3.1. Authorship identification

Authorship identification is an important problem to tackle, because many tasks need to recognize the author of a specific text. For example, finding out which author wrote an anonymous book or identifying the notes of a serial killer.

In this section, the challenge of verifying an author in a more realistic environment is approached. The utilized datasets (two of them) comprise one document by an unknown author and from one to five documents by a known author.

The datasets comprised subsets in various languages: English, Spanish, Dutch and Greek. The objective was to identify if an unknown text was written by the same author as the known texts.

It is important to bear in mind that this task becomes more difficult when the dataset comprises short documents, as the usual approaches cannot capture effective models with small amounts of words (Stamatatos, 2009). However, in actual cases in the forensic field, long texts seldom occur.

Various approaches have been proposed to produce more informative characteristics based on the style of the text. It is also possible to produce characteristics extracting lexical, syntactic or semantic information.

Lexical information usually limits word recounts and the appearance of common words. Syntactic information is able to obtain, at a certain extent, the words' context.

This work resorted to semantic information in order to find characteristics that help discriminate texts. For this purpose, a model using LDA was put forward. By using this method, the vocabulary of all the texts was considered at once and then, after a statistical process, the extent at which the relations between the words in each document take place was disclosed.

The approach proposed was assessed with two datasets, corpora PAN 2014 and 2015, on which in the task of verifying the author was performed.

5.3.2. Approaches to verify authorship

Several works have aimed at studying the challenge of author identification by means of generating various types of characteristics (Madigan *et al.*, 2005; Narayanan *et al.*, 2012).

The nature of the dataset sets the difficulty of the task; this is to say, how difficult it will be to extract suitable characteristics (Afroz *et al.*, 2005; Green *et al.*, 2013). Nirkhi and Dharaskar (2013) show by using a number of techniques that even if the number of authors increases and the size of the training dataset decreases, the performance of classification decreases. This seems logic because the size of training data is lower, thus the identification of characteristics is affected.

Many studies analyze author identification by means of their writing style (Pateriya *et al.*, 2012; Pavelec *et al.*, 2007; Verhoeven and Daelemans, 2014).

For example, in Layton *et al.* (2013), the characteristics based on style are compared with BoW (Bag of Words). It is intended to find the texts' authors in the same domain, Twitter. To generate the vectors of characteristics, style markers such as characters, long words, blank spaces, punctuation, hyperlinks or parts of speech, among others, are included. The authors found that a style-based approach was more informative than a BoW method; however, their best results were obtained when considering two authors, so there was a decrease in precision when the number of authors increased. This suggests that depending on how large the training set is, there will be stylistic characteristics that help distinguish one author from another, but not any other author.

Stylistic characteristics can also be applied to other tasks. Bergsma *et al.* (2012) combine characteristics to solve two sorts of problems. They tried to obtain characteristics of style, BoW and syntactic, in order to classify native and non-native English-language writers, texts written for conferences or workshops (place classification), and texts written by male and female writers (gender classification). The dataset comprises texts that are scientific articles; this sort of texts is longer, unlike emails, tweets and other brief texts. Therefore, it may be supposed that this might have helped identify texts written by non-natives with promising accuracy. However, long texts do not ensure good results, since the classification by place and gender had low performance.

There a several purposes for authorship identification. For example, Bradley *et al.* (2008) tried to demonstrate that it is possible to find out which author wrote a non-published article (for a conference or a newspaper), only considering the works cited in the text. By using latent semantic analysis (LSA), Dumais (2004) proposes the creation of a matrix of terms and documents in which the possible authors are considered documents, while the cited authors, terms. LSA assumes that the words with a close meaning will appear in similar text parts. The results of Bradley *et al.* (2008) showed that the blind review process must be contested.

Another instance is the work by Castro and Castro and Lindauer (2012), which intended to find out if the identity of Twitter users can be discovered by means of their writing style. The authors focused on characteristics such as the shape and length of the words, frequency of characters and void words, among others. With an RLSC (Regularized

Least Square Classification) algorithm, the authors correctly classified 41% of the tweets.

Pimas *et al.* (2015) approach author verification by means of generating three types of characteristics: stylometric, grammatical and statistical; this study bases on the challenge of authorship verification in PAN 2015. In this work, the distribution of topics is also considered, albeit the authors advice against using it, because the dataset was composed of a mix of topics, which affected their results. On one side, a crossed-validation model (10 folds) performs well; on the other, the model showed overtraining using the specified testing and training sets.

5.3.3. Datasets

The approach proposed was tested on two datasets, which are included in PAN 2015. It is based on crossed-topic and crossed-genre documents, and PAN 2014, which comprises specific-domain documents.

Table V.21: Overview of the dataset for PAN 2015 author identification task

Language	Training			Testing			Genre
	Problems	# docs.	Word mean × doc.	Problems	# docs.	Word mean × doc.	
English	100	200	366	500	452	536	Mix-topic
Spanish	100	500	954	100	1000	946	Mix-topic/genre
Dutch	100	276	354	165	380	360	Mix-genre
Greek	100	100	678	100	500	756	Mix-topic

Table V.22: Overview of the dataset for PAN 2014 author identification task

Language	Training			Testing			Genre
	Problems	# docs.	Word mean × doc.	Problems	# docs.	Word mean × doc.	
Dutch	96	268	412.4	96	287	398.1	Essays
Dutch	100	202	112.3	100	202	116.3	Reviews
English	200	729	848.0	200	718	833.2	Essays
English	100	200	3 137.8	200	400	6 104.0	Novels
Greek	100	385	1 404.0	100	368	1 536.6	Articles
Spanish	100	600	1 135.6	100	600	1 121.4	Articles

5.3.3.1. PAN 2015 dataset

In order to run the experiments with our approach, the corpus proposed was used to identify the author of PAN 2015 (Stamatatos *et al.*, 2015).

This dataset comprises four subsets, each one written in various languages: English, Spanish, Dutch and Greek. The subsets have significant differences. The subset in English comprises dialogues from plays; Spanish, opinion articles from online newspapers, magazines, blogs and literary essays; Dutch, essays and reviews; Greek, opinion articles on politics, health, sports, among others.

The corpus also has a different number of documents per subset, as described in table V.21. Plus, each language has a series of difficulties to solve that are specifically defined in section 5.3.4.

Owing to its nature, the dataset focused on problems that need to capture more specific information on the authors' writing style. For example, let us suppose that we know a person who worked for a newspaper writing about sports; but then, such individual decides to become independent and spend their life writing horror novels. A

possible task, on the one side, may be to find out which articles belong to the former sportswriter and which others to different authors. In this case, the documents' vocabulary may describe the author; for example, their rate of use of n-grams, this sort of task is called crossed topic.

Furthermore, another possible task is to discover if a horror novel was written by the novelist on the basis of the sports articles he or she had written before. Consequently, there is a dramatic change in the documents' genre and topic; that is to say, the intersection between the documents' vocabularies would noticeably reduce. Because of this, it is called crossed-genre task.

5.3.3.2. PAN 2014 dataset

PAN 2014 (Stamatatos *et al.*, 2014), as PAN 2015, comprises documents written in four languages: Dutch, English, Greek and Spanish. However, PAN 2014 documents share the same domain, i.e., they do not contain genre mixes.

Each dataset has from one to up to five known documents and one unknown; the challenge is to find out if the unknown document was written by the same author of the known documents. An overview of PAN 2014 is presented in table V.22.

5.3.4. Method proposed

As shown in section 5.3.2, the characteristics based on stylistics, syntactic and lexical information consider each document separately. For example, it is possible to obtain stylistic characteristics counting the number of void words and characters per document. These symbols,

in short texts of crossed domains, may occur at random and do not allow for capturing the writing style.

In a collection of crossed-domain texts, we infer that the linked words and their distribution in the texts can offer more informative characteristics, since the estimated topic distribution for LDA will depend on each text in the collection.

5.3.4.1. Method to verify authorship

There are various research works which have resorted to LDA as a source of characteristics. For example, Pacheco *et al.* (2015) faced the challenge of author verification in PAN 2015 by proposing a schema based on a universal background model (Reynolds *et al.*, 2000). By means of this schema, three vectors of characteristics were created: one for the author; another for each set of known documents; and, another for each unknown document. The three vectors were codified to produce a new one. Then the characteristics (including those generated by LDA) were selected by means of a random forest that hierarchically defines the importance of each characteristic.

In like manner, Moreau *et al.* (2015) resorted to a genetic algorithm to select the best sources of characteristics (strategies) to verify the author of PAN 2015. One of the strategies was LDA with which two five-topic vectors were generated by the means of the following steps: first, the authors divided documents into n-gram characters; second, a five-topic set was explored for the text of each author; finally, a five-topic set was sought to verify each document (unknown document). With the two vectors, the authors obtained a measure that indicates how

similar the vectors are regarding style; such measure is a characteristic of the genetic algorithm.

Savoy (2013), in a similar manner as the present book, uses the distribution of topics calculated by LDA. The challenge is to find which author, out of a set of them, wrote a specific document. Kullback-Leibler symmetric distance was calculated between the distribution of the unknown document and that of the documents by the authors. Then, the author was assigned in relation to the closest distance. This task is called authorship attribution since the challenge is to identify the authors of anonymous texts.

Authorship attribution was also studied by Seroussi *et al.* (2014); in their research topics, distributions were seen as vectors of characteristics (patterns) that were directly classified.

The main difference from the other works mentioned earlier is that in the present one, characteristics were generated using LDA under the supposition that a specific author has a specific way to relate words, and so, such author produces a text. This supposition led to analyze words that keep a relation in a written text and the extent to which the set of related words is present in the text. With this information, we produce subtractions of vectors to provide new vectors to generate a training dataset; this process is described below:

We intend to use LDA specifically to extract semantic information from the corpus. As previously mentioned, on the one side, in any given collection of texts, LDA can find relations according to the way they are used in the text; on the other, common stylistic approaches use rates of symbols in the texts to distinguish two documents written by different authors. As previously stated in section 5.3.2, as texts

shorten, the number of symbols tends to be insufficient to produce effective discriminating characteristics; this worsens when the number of authors increases.

We infer that writers have different ways to link words owing to the fact that each writer uses common phrases. Moreover, they use phrases at different rates and so, the written texts kept a background structure. For example, the authors usually add the phrase "the data gathered in the study suggest that...", unlike another author that resorts to "data seem to suggest that...". Therefore, words such as "the, in, to, that" may be included in various topics since, unlike LSA (Dumais, 2004), LDA can assign the same word to various topics in order to improve the management of polysemy. As a result, the use of these words in varying numbers will produce several topic distributions for each document.

To represent each problem, all the documents in the dataset are processed with LDA. Then, vectors that represent known and unknown documents are obtained (with real values, each topic's probability).

On the basis of a specific problem, we subtracted the vector of each known document from the vector of the unknown document (let us remember there is only one unknown document per problem; however, there are from one to five known documents).

It was found that the conversion of real values into 0, 1 did not significantly change the final results, this way we converted them into their binary values using arithmetic mean as a threshold. 0 represents the absence of the topic and 1 represents its presence. Therefore, the subtraction of vectors produced two possible values: 0 when the topics are the same and 1 when topics are different (figure V.3).

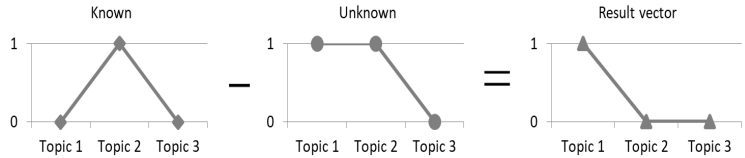


Figure V.3: Example of subtraction of the known document vector and the unknown document vector

5.3.5. Results

In the following experiments the documents were classified by means of an NB algorithm; moreover, various n-topics were specified for LDA. Therefore, patterns of n characteristics were produced for each document.

It was found that varying the number of topics also changed the performance of classification. As well, there is no *a priori* method to define how many topics must be chosen to increase performance; this way, an interval had to be empirically sought up to reaching the best results.

Since LDA is a stochastic method, the result obtained for each experiment can be different, so the average of 100 experiments is shown for each number of topics assessed; plus, standard deviation is included.

Let us keep in mind that the average of 100 experiments was calculated for each measure independently; that is to say, the product of accuracy and ROC area, in the corresponding tables, will not be proportional to FS measure (final score).

FS measure is the product of two values: c@1 (Reynolds *et al.*, 2000) and the area under ROC curve (Fawcett, 2006). The former is

an extension of accuracy, while the latter is a classification-performance measure that provides more complete results than accuracy.

Assuming that in a real case the testing set is unknown, firstly a five-fold cross validation of the training set was carried out to find the optimal interval of topics; secondly, the interval was applied to evaluate the datasets.

Table V.23: Performance in English, adjusting the number of topics

Topics	Binary values				Original values			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.743	0.733	0.554	0.129	0.495	0.407	0.202	0.019
3	**0.816**	0.853	0.697	0.041	0.672	0.699	0.474	0.090
4	0.800	**0.847**	**0.679**	0.049	0.633	0.637	0.407	0.080
5	0.781	0.835	0.654	0.067	0.631	0.646	0.412	0.085
6	0.759	0.811	0.618	0.075	0.622	0.621	0.389	0.064
7	0.778	0.790	0.586	0.084	0.595	0.598	0.358	0.064
8	0.746	0.800	0.599	0.064	0.591	0.594	0.356	0.083
9	0.735	0.787	0.581	0.075	0.616	0.624	0.388	0.073
10	0.724	0.767	0.559	0.084	0.604	0.603	0.367	0.067

Table V.24: Performance in Spanish, adjusting the number of topics

Topics	Binary values				Original values			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.633	0.632	0.402	0.037	0.637	0.672	0.428	0.032
3	0.730	0.765	0.561	0.080	0.673	0.702	0.473	0.046
4	**0.750**	**0.783**	**0.589**	0.066	0.678	0.697	0.474	0.053
5	0.740	0.776	0.576	0.071	0.678	0.698	0.475	0.047
6	0.751	0.777	0.586	0.072	0.661	0.694	0.463	0.061
7	0.754	0.777	0.589	0.075	0.664	0.697	0.465	0.068
8	0.726	0.756	0.551	0.072	0.646	0.676	0.439	0.064
9	0.719	0.747	0.540	0.079	0.639	0.667	0.429	0.070
10	0.715	0.738	0.530	0.075	0.635	0.658	0.420	0.066

Table V.25: Performance in Dutch, adjusting the number of topics

Topics	Binary values				Original values			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.722	0.721	0.521	0.030	0.682	0.703	0.479	0.023
3	**0.741**	**0.751**	**0.557**	0.030	0.685	0.710	0.487	0.029
4	0.717	0.736	0.530	0.067	0.671	0.707	0.475	0.035
5	0.675	0.730	0.496	0.067	0.636	0.676	0.432	0.050
6	0.668	0.724	0.485	0.049	0.617	0.661	0.409	0.044
7	0.681	0.730	0.498	0.043	0.619	0.651	0.404	0.046
8	0.690	0.730	0.505	0.041	0.626	0.653	0.410	0.045
9	0.696	0.727	0.508	0.050	0.633	0.660	0.419	0.040
10	0.690	0.715	0.494	0.045	0.629	0.657	0.414	0.038

Table V.26: Performance in Greek, adjusting the number of topics

Topics	Binary values				Original values			
	c@1	ROC	FS	FS-SD	c@1	ROC	FS	FS-SD
2	0.616	0.616	0.384	0.079	0.594	0.611	0.364	0.054
3	0.643	0.674	0.440	0.090	0.564	0.591	0.336	0.060
4	0.668	0.695	0.469	0.087	0.564	0.592	0.336	0.053
5	0.665	0.696	0.466	0.077	0.561	0.599	0.338	0.059
6	**0.671**	**0.709**	**0.479**	0.086	0.567	0.600	0.344	0.075
7	0.661	0.696	0.468	0.114	0.566	0.601	0.344	0.074
8	0.664	0.703	0.471	0.089	0.561	0.595	0.337	0.066
9	0.654	0.690	0.455	0.084	0.585	0.603	0.356	0.071
10	0.669	0.703	0.476	0.100	0.593	0.612	0.366	0.071

5.3.6. Results of PAN 2015 classification

Tables tables V.23-V.26 show a thorough analysis of the classification performance in each language in the identification of the author of PAN 2015. As noticed, it is possible to reach the best performance configuring various topics. For example, the method accomplishes the best result in English when four topics are used (table V.23). Each

table shows accuracy, ROC area, FS, FS-SD (final score – standard deviation). Besides, a comparison between the performance with original values and with LDA binary values is shown; such comparison suggests that for this study's established framework binary values produce a better performance in classification.

It is interesting to notice that with the vectors for few topics (1 up to 10), we can obtain an accuracy over 64%. In fact, it may be supposed, on one side, that the documents may have been categorized by topic, however, this supposition is unlikely, since the dataset utilized comprises a mix of subjects, as shown in section 5.3.3.1.

On the other, two experiments were carried out with a view to finding out if two documents written by the same author will be similar according to their topic distribution. Figure V.4 shows the addition of all the differences by topic in the testing dataset for English; as observed, the number of differences is large when the texts are written by different authors. Figure V.5 shows the same differences for Spanish.

The dataset without preprocessing was classified and the values displayed in table V.27 were found. The table shows the values corresponding to the experiment closest to each language's average.

According to table V.27, the best result was obtained for the English subset with 85.6% accuracy, even if the testing dataset is the largest in the corpus (500 problems). The Spanish subset holds the second place with 76.0% accuracy; the Dutch subset reached 70.9%; and, Greek, 64.0% accuracy.

We had the best and worst performances for English and Greek, respectively. Therefore, we cannot infer that the configuration of several topics made a difference in the results, as in reality both subsets are the

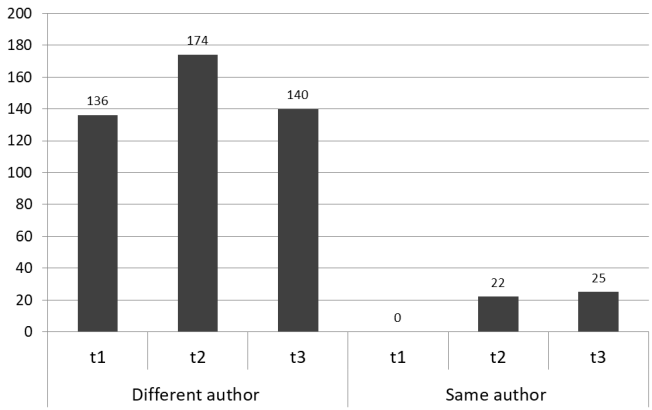


Figure V.4: Topic differences between documents written by the same or a different author (English subset)

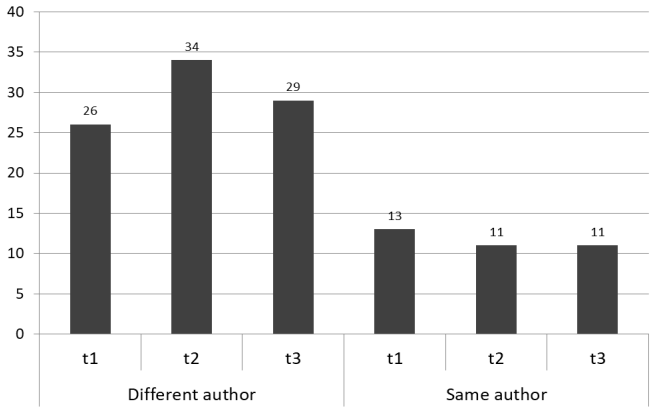


Figure V.5: Topic differences between documents written by the same or a different author (Spanish subset)

same type and one of them was not affected. In like manner, both for the Spanish and Dutch subsets (second and third place, respectively), the results do not lead to the conclusion that the mix of genres has some correlation with them.

The results obtained in this study were compared with those of the identification of author in PAN 2015 (Verhoeven and Daelemans, 2014). Therefore, the same performance measures as the authors of PAN 2015 task were utilized.

Table V.28 presents the best results for each language subset of the participants of PAN 2015 task; plus, the lowest final scores (FS) of the PAN 2015 participants are showed as well.

According to these results, our method seems to work better both for English and Dutch. Furthermore, the method proposed is capable of overcoming the FS results regarding the English subset and has better performance than Bartoli *et al.* (2015), and Bagnall (2015) for Dutch.

On the contrary, for the Spanish and Greek subsets the method proposed did not show good performance; however, our results surpass the worst FS scores of PAN 2015 participants and the results of ROC curve indicate that predictions are acceptable.

We exclude that bad results may be due to the amount of vocabulary, since 1 000 and 600 types of documents were obtained in Spanish and Greek, respectively; and only about 300 and 200 types for English and Dutch. Thereby, if performance depended on vocabulary, better results would be obtained in Spanish and Greek.

We infer that the results depend on the document sources; that is to say, English texts were taken from plays, which are dialogues; consequently, such documents may have a certain latent similar structure, as all of them were written for a specific purpose. Unlike documents in English, the others have various purposes; for example, the documents in Spanish were taken from newspapers and even personal *blogs*.

Table V.27: Results of each subset classification. Accuracy, precision (P), recall (R) and F-measure (F)

Dataset	Type	Accuracy	P	R	F
English	Cross-topic	85.6	0.864	0.856	0.855
Spanish	Cross-topic/genre	76.0	0.760	0.760	0.760
Dutch	Cross-genre	70.9	0.733	0.709	0.702
Greek	Cross-topic	64.0	0.646	0.640	0.640

Table V.28: Comparison of results with other authors'. FS=c@1*AUC

Author	Measure	Subsets			
		English	Spanish	Dutch	Greek
Bagnall (2015)	c@1	0.757	0.814	0.644	0.851
	AUC	0.811	0.886	0.700	0.882
	FS	0.614	0.721	0.451	0.750
Bartoli *et al.* (2015)	c@1	0.559	0.830	0.689	0.657
	AUC	0.578	0.932	0.751	0.698
	FS	0.323	0.773	0.518	0.458
Moreau *et al.* (2015)	c@1	0.638	0.755	0.770	0.781
	AUC	0.709	0.853	0.825	0.887
	FS	0.453	0.661	0.635	0.693
This study	c@1	0.816	0.750	0.741	0.671
	AUC	0.853	0.783	0.751	0.709
	FS	0.697	0.589	0.557	0.479
PAN'15 worst result	FS	0.201	0.095	0.089	0.212

5.3.7. Results of PAN 2014

The PAN 2014 task is the same as PAN 2015, previously explained; though, there is one main difference regarding the dataset, in PAN 2014 there are no mixed domains. Likewise, it has six subsets, and each comprises texts of the same genre (for instance, essays, articles, novels).

Table V.29: Details of the results from genre classification in which baseline was reached

Topics	Opinions-Dutch		Novels-English		Essays-English	
	FS	FS-DS	FS	FS-DS	FS	FS-DS
10	0.315	0.041	**0.391**	0.062	0.517	0.042
20	0.312	0.058	0.358	0.051	0.533	0.044
30	**0.325**	0.054	0.333	0.030	0.539	0.040
40	0.301	0.048	0.327	0.054	0.535	0.040
50	0.310	0.066	0.296	0.045	0.540	0.043
60	0.283	0.052	0.292	0.041	**0.551**	0.039
70	0.261	0.050	0.279	0.041	0.540	0.032
80	0.256	0.052	0.282	0.043	0.535	0.024
90	0.281	0.046	0.270	0.038	0.536	0.028
100	0.296	0.051	0.268	0.048	0.546	0.034

Table V.30: Details of the results from the genre classification in which baseline was not reached

Topics	Opinions-Dutch		Novels-English		Essays-English	
	FS	FS-DS	FS	FS-DS	FS	FS-DS
10	0.661	0.052	**0.436**	0.050	**0.322**	0.047
20	0.667	0.045	0.406	0.064	0.310	0.056
30	**0.648**	0.053	0.360	0.046	0.280	0.050
40	0.628	0.053	0.352	0.060	0.277	0.054
50	0.595	0.037	0.339	0.042	0.280	0.044
60	0.565	0.059	0.329	0.058	0.261	0.046
70	0.548	0.055	0.304	0.052	0.255	0.045
80	0.544	0.064	0.298	0.039	0.259	0.044
90	0.494	0.057	0.288	0.051	0.275	0.061
100	0.475	0.053	0.267	0.045	0.261	0.047

Table V.31: Comparison of performance, this work and PAN 2014 baseline

Genre	c@1	ROC area	This work's FS	PAN 2014 baseline	FS
Dutch-essays	0.778	0.855	0.667		0.685
Dutch-opinions	0.560	0.577	0.325		0.322
English-essays	0.705	0.780	0.551		0.288
English-novels	0.605	0.644	0.391		0.202
Greek-articles	0.631	0.686	0.436		0.452
Spanish-articles	0.555	0.576	0.322		0.378

Table V.32: Comparison of performance, this study's and PAN 2014 participants

Genre	PAN'14 best score	PAN'14 worst score	This study
Dutch-essays	0.823	0.307	0.667
Dutch-opinions	0.525	0.170	0.325
English-essays	0.513	0.270	0.551
English-novels	0.508	0.225	0.391
Greek-articles	0.720	0.281	0.436
Spanish-articles	0.698	0.248	0.322

In this section, the results of PAN 2014 classification are shown. Tables tables V.29 and V.30 show the details of the subsets' results for which PAN 2014 baseline was reached and for those that was not, respectively. In this case, baseline was surpassed, established by PAN 2014 challenge, for Dutch reviews, English novels and essays (three of the six subsets, see table V.31).

Table V.32 shows the best results of participants in PAN 2014. Although our method only surpasses the classification of English essays regarding the best scores, it is not below the worst ones.

Conclusion

Accordingly, this book showed the application of a continuous semantic space to mainly analyze the identification of deceptive text; in like manner, three study cases were presented; in them, the best results were obtained classifying documents from a single domain or topic. As well, these results were obtained using LDA. In the rest of the cases (mixed and crossed domains), the methods to generate characteristics with the best results were varied.

It has been demonstrated that the characteristics generated by means of a model of continuous semantic space have complemented characteristics generated by other approaches to a different extent; even though, it has been verified that when a text is written to deceive someone, the way words are used tends to change. Moreover, it was demonstrated that the words that represented each class (truthful and deceptive texts) will not be strictly definitive for a particular class in all the study cases.

This may suggest that it is best to opt for a method that accounts for the documents' representative characteristics, instead of one that looks for universal words to detect deception; or conversely, applying learning algorithms to an enormous dataset and find a set of words or linguistic phenomena that strongly discriminate one class or another.

However, tagging deceptive and truthful text is a rhetorical task; this way, research, thus far, is restricted to studies on a specific domain. Usually, most of the works on deception detection uses and combines various methods to generate characteristics and improve thus such detection; albeit, they disregard the fact that characteristics may change according to the nature of the text.

References

Afroz, Sadia, Michael Brennan and Rachel Greenstadt (2012).“Detecting hoaxes, frauds, and deception in writing style online” in *2012 IEEE Symposium on Security and Privacy, IEEE*, pp. 461-475.

Almela, Ángela, Rafael Valencia García and Pascual Cantos (2012). “Seeing through deception: A computational approach to deceit detection in written communication” in *Proceedings of the workshop on computational approaches to deception detection, Association for Computational Linguistics*, pp.15-22.

Bagnall, Douglas (2015). “Author identification using multi-headed recurrent neural networks”. Available in https://arxiv.org/ftp/arxiv/papers/1506/1506.04891.pdf, accessed April 2020.

Bartoli, Alberto *et al.* (2015). “An author verification approach based on differential features” in *Conference and Labs of the Evaluation forum*, vol. 1391, CEUR.

Bergsma, Shane, Matt Post and David Yarowsky (2012). “Stylometric analysis of scientific articles” in *Proceedings of the 2012 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 327-337. Available in https://www.aclweb.org/anthology/N12-1033.pdf, accessed April 2020.

Blei, David M., Andrew Y. Ng and Michael I. Jordan (2003), "Latent dirichlet allocation" in *Journal of Machine Learning Research*, vol. 3, pp. 993-1022.

Bradley, Joseph K., Patrick Gage Kelley and Aaron Roth (2008). "Author identification from citations". Available in http:www.cs.cmu.edu epxingClass1070106fproject-reportsbradley_kelley_roth.pdf, accessed April 2020.

Castro, Antonio and Brian Lindauer (2012). "Author identification on twitter". Available in http://cs229.stanford.edu/proj2012/CastroLindauer-AuthorIdentificationOnTwitter.pdf, accessed May 2020.

Chomsky, Noam (1956). "Three models for the description of language" in *IRE Transactions on information theory*, vol. 2, pp. 113-124.

DePaulo, Bella M. *et al.* (2003). "Cues to deception" in *Psychological bulletin*, vol. 129, p. 74.

Douglas, Dan and Kathleen M. Broussard (2000). "Longman grammar of spoken and written english" in *Tesol Quarterly*, vol. 34, pp. 787-788.

Dumais, Susan T. (2004), "Latent semantic analysis" in *Annual Review of Information Science and Technology*, vol. 38, pp. 188-230.

Ekman, Paul (1989). "Why lies fail and what behaviors betray a lie" in *Credibility assessment*, Springer, pp. 71-81.

Ekman, Paul (2009). Telling lies: Clues to deceit in the marketplace, politics, and marriage (revised edition), WW Norton & Company, New York.

Fawcett, Tom (2006). "An introduction to roc analysis" in *Pattern recognition letters*, vol. 27, pp. 861-874.

Fellbaum, Christiane (1998). "Towards a representation of idioms in WordNet" in *Proceedings of the workshop on the Use of WordNet in Natural Language Processing Systems* (Coling-ACL 1998), Montreal, Canada.

Feng, Song, Ritwik Banerjee and Yejin Choi (2012). "Syntactic stylometry for deception detection" in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Association for Computational Linguistics*, vol. 2, pp. 171-175.

Fornaciari, Tommaso and Massimo Poesio (2012). "On the use of homogenous sets of subjects in deceptive language analysis" in *Proceedings of the Workshop on Computational Approaches to Deception Detection, Association for Computational Linguistics*, pp. 39-47.

Fornaciari, Tommaso and Massimo Poesio (2014). "Identifying fake amazon reviews as learning from crowds" in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 279-287.

Fusilier, Donato Hernández *et al.* (2015). "Detection of opinion spam with character n-grams" in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, pp. 285-294.

Gokhman, Stephanie *et al.* (2012). "In search of a gold standard in studies of deception" in *Proceedings of the Workshop on Computational Approaches to Deception Detection, Association for Computational Linguistics*, pp. 23-30.

Green, Rachel M. and John W. Sheppard (2013). "Comparing frequency-and style-based features for twitter author identification" in *The Twenty-Sixth International FLAIRS Conference*. Available in https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5917/6043, accessed May 2020.

Guyon, Isabelle and André Elisseeff (2003). "An introduction to variable and feature selection" in *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182.

Hall, Mark Andrew (1999). "Correlation-based feature selection for machine learning". Available in https://www.cs.waikato.ac.nz/ mhall/thesis.pdf, accessed February 2020.

Hall, Mark *et al.* (2009), "The weka data mining software: an update" in *ACM SIGKDD explorations newsletter*, vol. 11, pp. 10-18.

Hauch, Valerie *et al.* (2012). "Linguistic cues to deception assessed by computer programs: A meta-analysis" in *Proceedings of the workshop on computational approaches to deception detection, Association for Computational Linguistics*, pp. 1-4.

Hernández Castañeda, Ángel and Hiram Calvo (2017). "Deceptive text detection using continuous semantic space models" in *Intelligent Data Analysis*, vol. 21, pp. 679-695.

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews" in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 168-177.

Indurkhya, Nitin and Fred J. Damerau (2010). *Handbook of natural language processing*, vol. 2. CRC Press.

Kamp, Hans (1981). "A theory of truth and semantic representation" in *Formal semantics -the essential readings*, pp. 189-222.

Keila, P. S. and David B. Skillicorn (2005). "Detecting unusual email communication" in *Proceedings of the 2005 conference of the Centre for Advanced Studies on Collaborative research*, IBM Press, pp. 117-125

Layton, Robert, Paul Watters and Richard Dazeley (2013). "Local ngrams for author identification" in *Notebook for PAN at CLEF 2013*. Available in https:pan.webis.dedownloadspublicationspaperslayton_2013.pdf, accessed February 2020.

Madigan, David *et al.* (2005). "Author identification on the large scale" in *Proceedings of the Meeting of the Classification Society of North America*, vol. 13.

Marneffe de, Marie-Catherine *et al.* (2006). "Generating typed dependency parses from phrase structure parses" in *Lrec*, vol. 6, pp. 449-454.

Masip, Jaume *et al.* (2012). "Deception detection from written accounts" in *Scandinavian Journal of Psychology*, vol. 53, pp. 103-111.

Mihalcea, Rada and Carlo Strapparava (2009). "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*, pp. 309-312.

Mohtasseb, Haytham, UK Lincoln and Amr Ahmed (2009). "More blogging features for author identification". Available in file:///C:/Users/Usuario/Downloads/MoreBloggingFeaturesforAuthor Identification.pdf, accessed February 2020.

Moreau, Erwan *et al.* (2015). "Author verification: basic stacked generalization applied to predictions from a set of heterogeneous learners-notebook for pan at clef 2015" in *CLEF 2015-Conference and Labs of the Evaluation forum*, CEUR.

Narayanan, Arvind *et al.* (2012). "On the feasibility of internet-scale author identification" in *2012 IEEE Symposium on Security and Privacy*, IEEE, pp. 300-314.

Newman, Matthew L. *et al.* (2003). "Lying words: Predicting deception from linguistic styles" in *Personality and social psychology bulletin*, vol. 29, pp. 665-675.

Nirkhi, Smita and Rajiv V. Dharaskar (2013). "Comparative study of authorship identification techniques for cyber forensics analysis" in *International Journal of Advanced Computer Science and Aplicattions*, vol. 14, núm. 5, pp. 32-35

Ott, Myle, Claire Cardie and Jeffrey T Hancock (2013), "Negative deceptive opinion spam" in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 497-501.

Ott, Myle *et al.* (2011). "Finding deceptive opinion spam by any stretch of the imagination" in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 309-319.

Pacheco, María Leonor, Kelwin Fernandes and Aldo Porco (2015). "Random forest with increased generalization: A universal background approach for authorship verification" in *CLEF* (Working Notes).

Pang, Bo, Lillian Lee and Shivakumar Vaithyanathan (2002). "Thumbs up?: sentiment classification using machine learning techniques" in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Association for Computational Linguistics*, vol. 10, pp. 79-86.

Pateriya, Pushpendra Kumar *et al.* (2012). "A study on author identification through stylometry" in *International Journal of Computer Science & Communication Networks*, vol. 2, p. 653.

Pavelec, Daniel, Edson Justino and Luiz S. Oliveira (2007). "Author identification using stylometric features" in *Inteligencia Artificial*. Revista Iberoamericana de Inteligencia Artificial, vol. 11.

Pennebaker, James W. *et al.* (2015), "The development and psychometric properties of liwc2015". *Technical report*. Available in file:C:UsersUsuarioDownloadsLIWC2015_LanguageManual.pdf, accessed February 2020.

Pérez Rosas, Verónica and Rada Mihalcea (2014a). "Cross-cultural deception detection" in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), vol. 2, pp. 440-445.

Pérez Rosas, Verónica and Rada Mihalcea (2014b), "Gender differences in deceivers writing style" in *Mexican International Conference on Artificial Intelligence*, Springer, pp. 163-174.

Pérez Rosas, Verónica *et al.* (2015). "Verbal and nonverbal clues for real-life deception detection" in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2336-2346.

Pimas, Oliver, Mark Kröll and Roman Kern (2015). "Know-center at pan 2015 author identification" in *Working Notes Papers of the CLEF*.

Pustejovsky, James (1991). "The generative lexicon" in *Computational Linguistics*, vol. 17, pp. 409-441.

Raykar, Vikas C., *et al.* (2010). "Learning from crowds" in *Journal of Machine Learning Research*, vol. 11, pp. 1297-1322.

Rayson, Paul, Andrew Wilson and Geoffrey Leech (2002). "Grammatical word class variation within the british national corpus sampler" in *New Frontiers of Corpus Research*, Brill Rodopi, pp. 295-306.

Reynolds, Douglas A., Thomas F Quatieri and Robert B. Dunn (2000). "Speaker verification using adapted gaussian mixture models" in *Digital signal processing*, vol. 10, pp. 19-41.

Savoy, Jacques (2013). "Authorship attribution based on a probabilistic topic model" in *Information Processing & Management*, vol. 49, pp. 341-354.

Schelleman Offermans, Karen and Harald Merckelbach (2010). "Fantasy proneness as a confounder of verbal lie detection tools" in *Journal of Investigative Psychology and Offender Profiling*, vol. 7, pp. 247-260.

Seroussi, Yanir, Ingrid Zukerman and Fabian Bohnert (2014). "Authorship attribution with topic models" in *Computational Linguistics*, vol. 40, pp. 269-310.

Sidorov, Grigori *et al.* (2014). "Syntactic n-grams as machine learning features for natural language processing" in *Expert Systems with Applications*, vol. 41, pp. 853-860.

Stamatatos, Efstathios (2009). "A survey of modern authorship attribution methods" in *Journal of the American Society for information Science and Technology*, vol. 60, pp. 538-556.

Stamatatos, Efstathios *et al.* (2015). "Overview of the pan/clef 2015 evaluation lab" in *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, pp. 518-538,

Stamatatos, Efstathios *et al.* (2014). "Overview of the author identification task at pan 2014" in *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers*, Sheffield, UK, 2014, pp. 1-21.

Toma, Catalina L. and Jeffrey T. Hancock (2012). "What lies beneath: The linguistic traces of deception in online dating profiles" in *Journal of Communication*, vol. 62, pp. 78-97.

Twitchell, Douglas P, Jay F. Nunamaker and Judee K. Burgoon (2004). "Using speech act profiling for deception detection" in *International Conference on Intelligence and Security Informatics*, Springer, pp. 403-410.

Verhoeven, Ben and Walter Daelemans (2014). "Clips stylometry investigation (CSI) corpus: a dutch corpus for the detection of age, gender, personality, sentiment and deception in text" in *LREC 2014-ninth International Conference on Language Resources and Evaluation*, pp. 3081-3085.

Vossen, Piek (1998). "Introduction to eurowordnet" in *EuroWordNet: A multilingual database with lexical semantic networks*, Springer, Dordrecht, pp. 1-17.

Vossen, Piek (2001). "Extending, trimming and fusing WordNet for technical documents" in *Proceedings on NAACL2001 workshop on WordNet and other lexical resources applications, extensions and customizations*, Pittsburgh, USA June 2001. The Association for Computational Linguistics.

Wierzbicka, Anna (1972). Semantic primitives, Frankfurt, Athenäum-Verl, serie: Linguistische Forschungen.

Williams, Shanna Mary *et al.* (2014). "Is the truth in your words? distinguishing children's deceptive and truthful statements" in *Journal of Criminology*, pp.1-9.

Xu, Qiongkai and Hai Zhao (2012). "Using deep linguistic features for finding deceptive opinion spam" in *Proceedings of Coling 2012: Posters*, pp. 1341-1350.

About the authors



Ángel Hernández Castañeda

Master and doctor in Computer Sciences from Center for Computer Research, National Polytechnic Institute, in 2013 and 2017, respectively. Author of a number of publications in high-impact international journals. Presently, he is a fulltime researcher-professor in the Autonomous University of the State of Mexico and a member of the National System of Researchers of Mexico. His research interests include natural language processing, data mining, machine learning and patter recognition.

Hiram Calvo

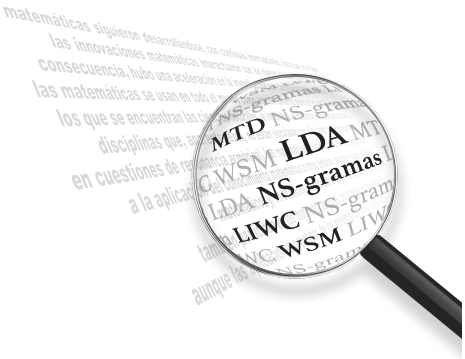
A holder of a master's degree in Computer Sciences and Engineering from the National Autonomous University of Mexico; then, he went on to study Computational Linguistics in the National Polytechnic Institute, where he was awarded the Lázaro Cárdenas Award. He made a post-doctoral stay in Japan, in Nara Institute of Science and Technology, in Dr. Yuji Matsumoto's laboratory. A member of the National System of Researchers of the National Council of Science and Technology, level II. At present, he is a fulltime researcher-professor in Center for Computer Research, National Polytechnic Institute, where he also acts as a coordinator of the Master in Computer Sciences.





René Arnulfo García Hernández

A graduate in Computer Systems Engineering from Technological Institute of Toluca, Mexico, in 2001; master in Computer Sciences from the National Center of Technological Research and Development, Mexico, 2003; doctor in Computer Sciences from the National Institute of Astrophysics, Optics and Electronics, Mexico, 2017. He is an author of more than 67 publications in the main international journals and congresses, as well as three books. He is an advisor in 38 graduate theses. Recognized as a national researcher, level II. Nowadays, he is a fulltime researcher-professor in UAEM Tianguistenco Professional Academic Unit.



Detección automática de texto engañoso mediante un modelo de espacio semántico continuo

Edición bilingüe

de la autoría de Ángel Hernández Castañeda, Hiram Calvo y René Arnulfo García Hernández se terminó de editar el 6 de noviembre de 2020.

Por disposición del Reglamento de Acceso Abierto de la Universidad Autónoma del Estado de México, se publica la versión PDF de este libro en el Repositorio Institucional de la Máxima Casa de Estudios estatal.

Una línea de estudio particular dentro de la inteligencia artificial es el procesamiento del lenguaje natural, el cual tiene como objetivo facilitar la comunicación entre la computadora y el lenguaje humano.

Por lo anterior, este libro presenta un nuevo método para identificar el engaño en los textos mediante técnicas del procesamiento del lenguaje natural y algoritmos de aprendizaje automático. El proceso general que sigue el enfoque propuesto es el siguiente: mediante herramientas del procesamiento del lenguaje natural, los textos se convierten a un espacio vectorial, es decir, se generan vectores de características; después son aplicados diversos métodos de selección de atributos para discriminar aquellas características poco relevantes; por último, los vectores resultantes se envían a un algoritmo de clasificación supervisada que identifique si son engañosos o veraces.