Universidad del Bío-Bío, Chile

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

# Modelo de Detección Automática de Ironía en Textos en Español

Tesis presentada por Marcelo Pinto Cruces.
para obtener el grado de Magíster en Ciencias de la Computación
Dirigida por: Alejandra Segura y Christian Vidal

2017

# Agradecimientos

Ante todo quiero agradecer a DIOS por su infinito amor.

Todo esto es gracias al apoyo de mis padres Marcelo y Cecilia, quienes desde siempre han sido los mejores padres que una persona podría tener. Gracias por su esfuerzo y dedicación de siempre. Quiero agradecer a mi amada esposa Pamela, por su amor, su apoyo, su paciencia y su inteligencia, por ser mi mejor amiga, y mi pilar fundamental. Gracias a mi familia, mi tía Mary, mis hermanos Francisco, Felipe, Patricia, Sebastián. A mis suegros Juana y Raúl.

A mis queridos profesores Christian Vidal y Alejandra Segura por su apoyo y confianza de siempre. A todos quienes me han apoyado en este trabajo compartiendo su conocimiento y consejos a lo largo de esta carrera: Jorge, Lorena, Joel, Hugo, Olivia, Daniel y Sebastián Luna y a mi amigo Adrián por su apoyo de toda una vida.

Gracias además a mi querida Universidad del Bío-Bío, mi alma mater.

Finalmente, dedico este trabajo a mi abuelita Florinda Alveal Gallardo, que hoy ve este logro desde un lugar muy feliz, lejos de aquí.

# Abstract

Nowadays, the automatic irony detection is an open topic of research, which is being approached by different research groups. Irony is defined in different ways, but most works converge in that it is the words use to express something different and opposite to the literal sense and given that people often use irony to express their opinions, the detection of this is part of the concern of researchers in the area of analysis of feelings, which uses the detection of irony to avoid misinterpretations of irony sentences as literal phrases. This paper proposes a new irony detection model for spanish texts based on the approaches studied in a systematic literature review, in which were reviewed articles related to the detection of irony, sarcasm and opinion mining. For evaluate our model we constructed a corpus of evaluation composed of spanish tweets searched according to different thematic domains, which were labeled as ironic and not ironic by human evaluators. In addition, was developed an application that allows to perform tasks of pre-processing and processing of each characteristic that composes our model to finally generate datasets for evaluation. Our model was based on the most used irony characteristics in the revised models, besides we introducing 2 new ones, being finally conformed by the attributes: Use of Emoticons, Upper-Case Letter, Punctuation Marks, Irony Typical Words, Text-Emoticon Contradiction Analysis, Ngrams and Skipgrams Analysis. We obtained general scores of up to 78 percent, a value considered acceptable in comparison to the existing models and promising considering the practically null existence of works of automatic detection of irony in texts in Spanish.

***Keywords*** — Irony Detection, Opinion Mining, Natural Language Processing.

# Resumen

Actualmente, la detección automática de ironía en textos es un tema abierto de trabajo, el cual está siendo abordado por distintos grupos de investigación. La ironía es definida de distintas formas y la mayoría de los trabajos converge en definirla como el uso de palabras para expresar algo distinto y opuesto al sentido literal. Además, dado que las personas a menudo usan la ironía para expresar sus opiniones, la detección de esta es de interés para investigadores del área de análisis de sentimientos, debido a que permite evitar interpretaciones erróneas de opiniones y decidir cuándo una de ellas es o no irónica o literal. Este trabajo propone un nuevo modelo de detección de ironía en textos en español a partir de los enfoques estudiados en una revisión sistemática de literatura, en la cual se revisaron artículos relacionados con la detección de ironía, sarcasmo y minería de opinión. Para evaluar el modelo se construyó un corpus de evaluación compuesto de tweets en español recopilados de acuerdo a distintos dominios temáticos, los cuales fueron etiquetados como irónicos y no irónicos por evaluadores humanos. Junto con lo anterior, se desarrolló una aplicación que permite realizar tareas de pre-procesado y procesado de cada característica que compone nuestro modelo para finalmente generar el dataset para evaluación. Este nuevo modelo tomó como base las características de ironía más utilizadas en los modelos revisados y además introduce 2 nuevas características. Finalmente, el modelo quedó conformado por los atributos: **Uso de emoticonos, Uso de Mayúsculas, Uso de Signos de Puntuación, Uso de Palabras típicas de ironía, Uso de Adverbios temporales y no temporales, Análisis de Contradicción Texto-Emoticón, Análisis de ngramas y Análisis de skipgramas**. Se obtuvieron rendimientos generales de hasta un 78 %, valor considerado aceptable en comparación a los modelos existentes y prometedor considerando la prácticamente nula existencia de trabajos de detección automática de ironía en textos en español.

***Palabras Clave*** — Detección Automática de Ironía, Minería de Opinión, Procesamiento de Lenguaje Natural

# Índice general

# Índice de figuras

# Índice de tablas

# Capítulo 1

# Introducción

Actualmente, la detección automática de ironía en textos es un tema abierto de investigación, el cual está siendo abordado por distintos investigadores relacionados, por ejemplo con disciplinas de Procesamiento de Lenguaje Natural (NLP) y Machine Learning (ML) (Wallace, 2015). Existen distintas corrientes de investigación dentro de la misma temática. Por un lado, algunos trabajos se enfocan en definir la ironía desde una perspectiva teórica, y por otro, existen trabajos dedicados al estudio y creación de modelos computacionales para detectar ironía y sarcasmo en textos (Wallace, 2015). Este trabajo se enmarca dentro de este último enfoque de investigación.

La ironía es definida de distintas formas, pero la mayoría de los trabajos converge en que es el uso de palabras para expresar algo distinto y opuesto al sentido literal (Kong y Qiu, 2011). La ironía es transversal a la pronunciación, elección léxica, la estructura sintáctica y semántica. Según Reyes et al. en (Reyes et al., 2013) aún no existe una solución general en un solo algoritmo o técnica que detecte la ironía en textos. Dado que las personas a menudo usan la ironía para expresar sus opiniones (Nagwanshi y Veni Madhavan, 2014), la detección de esta es parte de la preocupación de investigadores del área de análisis de sentimientos, que utiliza la detección de ironía para evitar interpretaciones erróneas de frases de ironía como frases literales.

La ironía comparte características importantes con el concepto de sarcasmo y se diferencian principalmente debido a que este último tiene una intención de burla y/o mala intención (Real Academia de la Lengua Española, RAE). Dado lo anterior, en este trabajo se incluirán trabajos de detección de sarcasmo, ya que muchas técnicas se pueden homologar y enriquecer el modelo de detección de ironía.

Entendemos por modelo de detección de ironía a un conjunto de características y técnicas que permiten decidir si un texto es o no irónico. Además, existen distintos enfoques en los cuales se basan estos modelos. Un enfoque es un conjunto de técnicas que siguen una lógica común, por ejemplo, enfoque de Machine Learning o enfoque de Análisis Lexicón. Finalmente, dependiendo de cada enfoque, existen técnicas que apoyan el funcionamiento de un modelo, por ejemplo un árbol de decisión en el enfoque Machine Learning.

Esta investigación propone un nuevo modelo de detección automática de ironía en textos escritos en español, a partir de los enfoques estudiados en una revisión sistemática de literatura (Kitchenham, 2004). La revisión sistemática estudiará modelos que tratan con la detección de

ironía, comparando sus rendimientos, alcances de idioma y aplicaciones.

Hasta antes de la realización de este trabajo no se habían encontrado modelos de detección de ironía que funcionen en lenguaje español, por lo tanto, este trabajo propone un modelo que pueda funcionar en el idioma español y que combine las técnicas y enfoques de los modelos existentes. Este modelo de detección de ironía será evaluado sobre un corpus de mensajes de Twitter.

Este documento se estructura de la siguiente manera: En el capítulo Objetivos e Hipótesis se formulan objetivos generales, específicos, metodología de investigación e hipótesis. El siguiente capítulo corresponde a la definición del protocolo de la revisión sistemática aplicada en este trabajo. Se exponen los principales conceptos relacionados a la ironía, minería de opinión y sarcasmo, para luego exponer el marco teórico en el siguiente capítulo. Una vez definido el marco teórico, se describe todo el proceso y resultados de la construcción del corpus de evaluación en el capítulo siguiente, para definir y describir el modelo de detección de ironía propuesto en este trabajo en el capítulo modelo de detección de ironía. Finalmente, se describe los experimentos realizados en los últimos capítulos para terminar con los trabajos futuros y la conclusión del estudio.

# Capítulo 2

# Objetivos e Hipótesis de Investigación

## 2.1. Hipótesis

Es posible proponer un modelo para la detección de ironía en textos en español, basado en enfoques propuestos.

## 2.2. Objetivos

### 2.2.1. Objetivo General

Proponer un modelo que detecte ironía en textos, en idioma español, a partir de propuestas que utilizan Machine Learning y de análisis lexicón.

### 2.2.2. Objetivos Específicos

- Realizar una revisión sistemática de literatura respecto a enfoques actuales de detección de ironía en textos, sus datasets, resultados y aplicaciones.
- Crear un Corpus de Evaluación formado por mensajes de Twitter en español.
- Identificar las características más relevantes de los modelos de detección de ironía encontrados en la revisión sistemática de literatura y seleccionar aquellas que serán incluidas en el nuevo modelo a proponer.
- Implementar el nuevo modelo de detección de ironía en un prototipo de aplicación desarrollada en una plataforma a definir.
- Validar la propuesta mediante un experimento de evaluación.
- Analizar de forma crítica y comparativa los resultados obtenidos y emitir conclusiones.

## 2.3. Alcance de la investigación

Considerando que ironía y sarcasmo no son diferenciados en forma precisa y a pesar que esta investigación estará centrada solo en la detección de ironía, también se estudiarán los trabajos que traten con detección sarcasmo, cuyos métodos, enfoques y técnicas puedan ser de utilidad en la detección de ironía.

Por otro lado, debido a que hasta la fecha no se han encontrado trabajos que propongan modelos de detección de ironía aplicados al idioma español, este trabajo estará enfocado en la detección de ironía en textos en español y el modelo a proponer será validado en un corpus de evaluación de mensajes de Twitter, de tal forma que se pueda comprobar si es o no posible aplicar las características de los modelos actuales al idioma español.

Finalmente, se espera que el resultado final de este trabajo de investigación sea un nuevo modelo de detección de ironía en textos en español basado en las características de los distintos enfoques existentes en el estado del arte aplicables al idioma español.

## 2.4. Metodología de Trabajo

Se realizarán las siguientes tareas alineadas a los objetivos específicos de esta investigación:

- Realizar una revisión sistemática de literatura donde se recopilen los enfoques actuales de detección de ironía en textos, sus datasets, resultados y aplicaciones.
- Crear un corpus mediante la recopilación de tweets que se llevará a cabo usando una aplicación basada en la API de Twitter. El corpus estará conformado por tweets escritos en idioma español y se guiará por una temática relevante y contingente.
- Clasificar mediante evaluadores, la recopilación de tweets candidatos en dos: criterios de irónico y no irónico. El grupo de evaluadores será conformado por un grupo de personas de con características y cantidad a definir.
- Proponer un modelo de detección de ironía en textos basado en las características más relevantes de los modelos encontrados en la revisión sistemática de literatura y seleccionar aquellas que serán incluidas.
- Implementar el nuevo modelo de detección de ironía en un prototipo de aplicación desarrollada en una plataforma a definir.
- Diseñar y planificar el experimento en el cual se realizará la evaluación del nuevo modelo propuesto utilizando el corpus etiquetado.
- Ejecutar el experimento de evaluación y registrar resultados obtenidos.
- Analizar de forma crítica y comparativa los resultados obtenidos, realizando gráficos y discusiones al respecto.

# Capítulo 3

# Marco Teórico

Este capítulo describe el marco teórico que guía esta investigación, el cual fue recopilado mediante una revisión sistemática de literatura. Se describe el proceso de revisión sistemática, la conceptualización de elementos escenciales en la detección de ironía y finalmente los principales modelos de detección de ironía y sarcasmos analizados en este trabajo.

## 3.1. Revisión Sistemática de Literatura

Se realizó una revisión sistemática de literatura con la finalidad de recopilar el estado del arte de la temática de esta investigación. Dicha revisión, consistió en el análisis de 46 textos, los cuales, fueron revisados e incorporados de acuerdo a filtros y criterios de selección. De acuerdo a lo expuesto en secciones anteriores, y a pesar que esta propuesta se acota solo a la detección de ironía, igual se contemplaron trabajos relacionados a la detección de sarcasmo, ya que en general utilizan enfoques y técnicas parecidas. A continuación, se detalla el protocolo seguido en la revisión sistemática y posteriormente sus resultados.

### 3.1.1. Preguntas de Investigación

Se formularon las siguientes preguntas de investigación que guían la revisión sistemática, orientadas a las diversas técnicas de detección de ironía o sarcasmo a buscar dentro de la literatura:

(a) ¿La técnica aborda sarcasmo, ironía o ambas?
(b) ¿Cuál es el grado de validación que tiene la técnica?
(c) ¿Qué desempeño tiene la técnica?
(d) ¿La técnica es aplicable a un idioma o más?
(e) ¿Qué nivel de integración tiene la herramienta y la técnica?
(f) ¿Qué enfoques se han usado para abordar la ironía o sarcasmo en análisis de sentimientos?
(g) ¿Cuál es el dominio de aplicación donde se aplica(ría) la técnica?

### 3.1.2. Protocolo de Búsqueda

Para realizar la búsqueda de literatura, se utilizaron las siguientes cadenas de búsqueda: *("Document Title": irony OR "Document Title":sarcasm) AND ("Abstract": "sentiment analysis" OR "Abstract":"Opinion mining" OR "Abstract":"affect analysis" OR "Abstract":"affective computing" OR "Abstract":"subjectivity analysis" OR "Abstract":sentic computing OR "Abstract":"irony detection")*. La estrategia de búsqueda fue basada en los siguientes aspectos:

- *En recursos con herramientas de búsqueda*: Ingresar de forma escalada los términos y las combinaciones entre ellos.
- *En Internet*: Utilizar Wos, Springer, IEEE, Google Scholars como motores de búsqueda, debido a que estos son los principales buscadores utilizados. Si uno o más documentos están inaccesibles, buscar en páginas y sitios alternativos.
- *Autores*: Identificar autores relevantes, acceder directamente a sus páginas personales para la búsqueda de material.
- *En artículos*: Detectar referencias bibliográficas de utilidad, en base a ellas buscar directamente el artículo citado usando los antecedentes que aparecen en la citación.

### 3.1.3. Protocolo de Revisión

El protocolo de revisión de literatura, constó de los siguientes aspectos:

(a) Normas de Revisión: Recopilar trabajos completos e impresos. Revisar críticamente introducción, resumen, conclusión y referencias. Decidir su inclusión/exclusión respecto a normas de este protocolo. Rotular cada artículo comentando la decisión y comentarios relevantes.

(b) Criterios de Inclusión: Se incluirán todos aquellos trabajos o estudios que aborden el tema de detección de ironía o sarcasmo en textos que estén disponibles en portales y/o en la web que se enmarquen dentro de los siguientes temas:
   - Planteamiento de modelos de detección de ironía o sarcasmo en textos.
   - Aplicaciones de detección de ironía o sarcasmo en textos aplicadas en casos reales.
   - Creación de corpus de evaluación para modelos de detección de ironía o sarcasmo en textos.
   - Desafíos y trabajos futuros relacionados a detección de ironía o sarcasmo en textos.

(c) Criterios de Exclusión: Se excluirán todos aquellos estudios que a pesar de contener los términos de búsqueda o combinación de ellos, no contienen información relevante sobre el tema y/o no abordan el tema de interés. En esta etapa se descartaron todos los estudios que no propusieran formalmente un modelo de detección de ironía ni describieran de forma clara el experimento de evaluación.

(d) Estrategia de Extracción de Datos: Por cada estudio seleccionado, se realizará una lectura crítica con el objeto de extraer datos para el trabajo. Primero se leerá la introducción, resumen, conclusión y referencias para saber:
   - *Introducción y Referencias*: A qué comunidad está dirigido el estudio.
   - *Resumen, Introducción y Conclusión*: Cuales son sus contribuciones y aportes.
   - *Resumen, Conclusión e Introducción*: Cuales son las consecuencias de sus contribu-

ciones y aportes y como se ven reflejados en aplicaciones de la vida real.
- *Cuerpo del Artículo*: Incluir, separar y analizar detalladamente la información útil en el estudio.
- *Cuerpo del Artículo*: Comprender los experimentos, el marco de trabajo sobre el cual fueron desarrollados.
- *Conclusión*: Observar y analizar trabajos futuros.

(e) Estrategia de Síntesis de Datos: Los datos serán resumidos de acuerdo a los siguientes temas:
- Estado del arte y avances de la detección de ironía o sarcasmo en textos.
- Modelo de detección de ironía o sarcasmo planteado.
- Corpus de evaluación utilizado y características del experimento.
- Tipo de enfoque utilizado (machine learning, análisis de lexicón, híbrido).
- Resultados del experimento.

### 3.1.4. Selección de Estudios Primarios

Se realizó la búsqueda en los portales Wos, Springer, IEEE, Google Scholars donde se recopilaron 46 artículos en inglés.

### 3.1.5. Selección de Estudios Secundarios

De los 46 estudios seleccionados en la etapa primaria, se procedió a leer en detalle de acuerdo al protocolo de revisión descrito cada paper. Además, se desecharon los trabajos que estuvieran orientados principalmente a la teoría de ironía o sarcamo, debido a que este trabajo está enfocado esencialmente al estudio de la ironía en términos computacionales, dejando de éstos solo los más relevantes para fines de definiciones formales. Finalmente, se seleccionaron 16 trabajos con los que se conformará sólo el marco teórico, es decir, los modelos de detección de ironía y sarcasmo. Cabe destacar que en este trabajo se incluyen numerosos artículos además de los mencionados en esta revisión sistemática, ya que los conceptos tratados en esta investigación son bastante amplios y se necesitan diversas fuentes para abordarlos de forma adecuada.

## 3.2. Conceptualización

La detección de ironía es una tarea compleja y desafiante dentro del marco del concepto de Minería de Opinión (Arora y Srinivasa, 2014). Se presentan a continuación algunos conceptos de importancia para la detección de ironía en textos, específicamente relacionados a la minería de opinión y a cómo definen la ironía la mayoría de los trabajos hasta ahora.

### 3.2.1. Minería de Opinión

Según (Arora y Srinivasa, 2014), los textos contienen conocimiento que puede ser clasificado en dos categorías:

- Hechos: Declaraciones típicamente objetivas acerca de una entidad o evento.

- Opiniones: Declaraciones que expresan sentimientos y sensaciones del autor acerca de entidades o eventos.

Estos autores proponen una clasificación de los aspectos que tiene la minería de opinión, los cuales definen la estructura que tiene una opinión, además de herramientas y tecnologías para la minería de opinión.

Mediante una opinión, un autor expresa sentimientos acerca de algo o de un aspecto de ese algo. Los tres elementos relevantes de una opinión son:

- Entidad Objetivo: Corresponde a una entidad, sobre la cual se emite una opinión acerca de ella.
- Autor: Es quien expresa la opinión acerca de una entidad objetivo.
- Sentimiento: Corresponde al sentimiento del autor por la entidad implicada. La polaridad de este sentimiento se clasifica en Positiva, Negativa o Neutral.

(Arora y Srinivasa, 2014) proponen distintas clasificaciones para las opiniones, algunas respecto a su polaridad y otras respecto a su categoría. Se detallan a continuación:

- Respecto a su polaridad: como positivas, negativas o neutrales.
- Respecto a su categoría: como directa, indirecta y comparativas.

En las Tablas 3.1 y 3.2 se presentan ejemplos de cada clasificación, los cuales consisten en oraciones representativas de cada clasificación:

| Clasificación | Ejemplo de Opinión |
|---|---|
| Positiva | "Alimentar perros vagos es una muy buena iniciativa" |
| Negativa | "Es demasiado complicado implementar un sistema de pagos automáticos en los taxis" |
| Neutral | "No apoyo ni me opongo al servicio Uber." |

Tabla 3.1: Ejemplo de Clasificación de Opinión de acuerdo a su Polaridad (Arora y Srinivasa, 2014).

| Clasificación | Ejemplo de Opinión |
|---|---|
| Directa | "El Rey León es una hermosa película" |
| Indirecta | "¡Qué mejor que una película para poder dormir!" |
| Comparativa | "La marca Colún es mejor que la Soprole." |

Tabla 3.2: Ejemplo de Clasificación de Opinión de acuerdo a su Categoría (Arora y Srinivasa, 2014).

En relación a las herramientas y tecnologías para la minería de opinión, los esfuerzos de investigación se concentran principalmente en extraer elementos de una opinión y agrupar los sentimientos de la opinión.

- Exploración de elementos de una opinión: Los métodos que se utilizan para explorar opiniones son: enfoque basado en reglas, machine learning y algoritmos estadísticos con lexicones.

Se entiende por lexicón a un diccionario de un idioma específico que contiene palabras con una polaridad pre-definida.

- Agregación y agrupamiento de sentimientos en una opinión: Para realizar este agrupamiento existen dos criterios: en primer lugar, el nivel de granularidad de cada opinión que debe extraerse del corpus y el nivel de agregación para el sentimiento general de la opinión. Los siguientes niveles para el análisis de opinión se escogen en función del dominio de la aplicación:
  - Análisis a nivel de Documento: Para realizar este análisis se debe cumplir que el documento contenga una opinión para una sola entidad y aspecto. La sumarización consiste en encontrar un sentimiento predominante en forma general en el documento.
  - Análisis a nivel de Oración o Sentencia: Posee un tratamiento similar al análisis nivel de documento, pero con la diferencia que acá se cumple que cada oración contiene una opinión para una entidad y aspecto, además que algunas oraciones no contienen una opinión, es decir, son objetivas.
  - Análisis a nivel de Aspecto: Este análisis considera que un documento contiene opiniones de muchas entidades y sus aspectos. Por lo tanto, se descubren todas las entidades, sus aspectos y sentimientos.

Los autores analizan además cómo funciona la minería de opinión. Cada corpus se reduce a "documentos", los cuales, luego de identificar su idioma se verifica si existe o no un Lexicón disponible. Una vez seleccionado un lexicón, se determina el nivel de granularidad del documento: nivel de oración, nivel de documento o nivel de aspecto. Ya realizado lo anterior, se procede a analizar las entidades presentes e inferir la polaridad para cada carácterística de la entidad. Cuando las opiniones se encuentran identificadas, se procede a detectar y marcar correos electrónicos o mensajes "basura", (SPAM) en ellas. Luego se realiza la detección de ironía, con la finalidad de verificar si la polaridad de cada opinión es realmente correcta o si se trata de un caso de ironía verbal. Finalmente, se agrupan las opiniones respecto a sus polaridades y/o clasificaciones. La Figura 3.1 muestra el funcionamiento de este ciclo de vida básico.

### 3.2.2. La Ironía

La ironía es una forma sofisticada de uso del lenguaje que reconoce una brecha entre el significado y el sentido literal de las palabras. Se puede dividir en dos grandes categorías: la ironía situacional y verbal (Forslid y Wikén, 2015). La primera es como, por ejemplo, una compañía de cigarrillos que tiene señalética de no fumar en la sala de compras. La segunda se define comúnmente como "diciendo lo contrario de lo que quiere decir", donde se supone que la diferencia entre el dicho y el significado de ser clara.

Ejemplos de ironía verbal podrían ser:

- "Que manera de reirme con Ricardo Meruane (?)"
- "Tembló? No importa!, La ONEMI nos cuida! jaja"

Este trabajo abordará la **ironía verbal**, para la cual, existen distintas definiciones en la literatura actual.

En (Wallace, 2015) se menciona una definición de ironía como "Un tropo retórico (sustitución

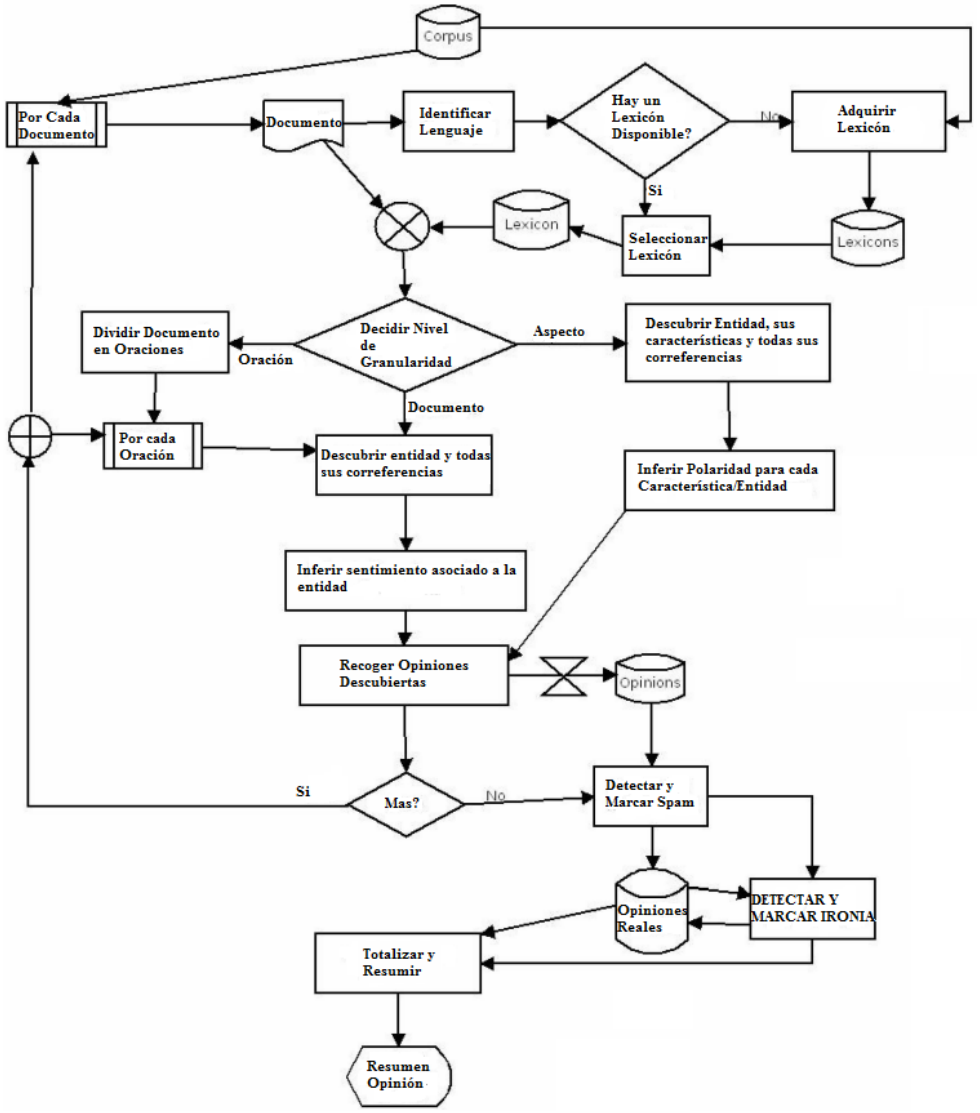


Figura 3.1: Ciclo de vida Básico de Minería de Opinión (Arora y Srinivasa, 2014).

de una expresión por otra cuyo sentido es figurado) en la cual un orador dice lo contrario a lo que quiere expresar". Esta definición es bastante similar a la de (Kong y Qiu, 2011) en la cual define como "el uso de palabras para expresar algo distinto y opuesto al sentido literal".

(Clark y Gerrig, 1984) propone dos teorías para explicar los fundamentos de un mensaje irónico. La primera es llamada "teoría de la mención, y plantea que la ironía puede ser explicada a través de la diferencia existente en el uso y la mención de cierta expresión. Un hablante irónico no utiliza las palabras de una frase como si fueran propias, si no que busca solo mencionarlas haciendo referencia a otras frases (que casi siempre son dichas por otra persona) las cuales pueden

formar parte del conocimiento popular u opiniones aceptadas, con el fin de hacer notar desprecio hacia estas o de indicar lo opuesto a lo que se dice. La segunda teoría corresponde a la "Teoría de la pretensión.[en] la que se plantea que un hablante al ser irónico, pretende ser reconocido o percibido de cierta forma (sin dar a conocer esta pretensión para que no se pierda el efecto en el oyente).

En definitiva, la ironía estudiada bajo cualquier enfoque (lingüístico, filosófico o psicológico), en lugar de llegar con una solución final, da origen a diferentes perspectivas del fenómeno, es decir, no existe un acuerdo de una definición final de ironía (Sperber y Wilson, 1995)

### 3.2.3. Detección de la ironía en Texto

(Gibbs, 2000) señala que existen muchas expresiones irónicas que no encajan dentro de las teorías de pretensión o de mención, y que la entonación del hablante al momento de expresarse juega un rol clave en la comprensión por parte de un oyente. Por otro lado, (Wilson y Sperber, 1992) sostiene que al intentar detectar ironía existe el riesgo de cometer un error, debido a que las intenciones del comunicador no pueden ser decodificadas, sino que deben ser inferidas. En vista de lo anterior y de lo expresado además por (Sperber y Wilson, 1995) se puede concluir que no se ha encontrado aún un método 100 % certero en la detección de ironía. Además, actualmente no existen métodos computacionales específicos para la detección de ironía en texto (Seerat y Azam, 2012) y, adicionalmente, no se han encontrado trabajos que traten la detección de ironía en textos en español.

Los modelos existentes para la detección de ironía y sarcasmo se basan en los enfoques principales de las técnicas de Detección de Sentimientos, los cuales son (Muhammad et al., 2015):

- Enfoque basado en Lexicón: Funcionan con una recopilación de términos y frases que corresponden con un concepto, en este caso, con ironía o sarcasmo. Esta recopilación es utilizada para comparar y buscar semejanza.
- Enfoque basado en Machine Learning: Utiliza algoritmos y enfoques que generan aprendizaje automático a partir de información a modo de entrenamiento. En el caso de la ironía y sarcasmo, los enfoques "aprenden" características lingüísticas y sintácticas definidas para cada caso.
- Enfoque Híbrido: Utiliza las técnicas anteriores combinando sus características y enfoques.

### 3.2.4. El sarcasmo

Actualmente, no existe un consenso respecto a las diferencias conceptuales entre ironía y sarcasmo. La RAE (Real Academia Española) define la ironía como "Expresión que da a entender algo contrario o diferente de lo que se dice, generalmente como burla disimulada" mientras que el sarcasmo es definido como "Burla sangrienta, ironía mordaz y cruel con que se ofende o maltrata a alguien o algo." El sarcasmo, a diferencia de la ironía, tiene la característica de ser burlesco y cruel, además de estar focalizado en la ofensa de algo o alguien. En esta tesis se incluirán los trabajos más relevantes relacionados a la detección de sarcasmo, debido a que la ironía y

sarcasmo comparten características similares y los diversos modelos de detección de sarcamo podrían complementar el modelo de detección de ironía que se propone en este trabajo.

## 3.3. Modelos de Detección de Ironía

A continuación se presentan los estudios seleccionados en la revisión sistemática que proponen modelos de detección de ironía en textos. Para cada uno de ellos, se muestra un resumen de características y conceptos.

Antonio Reyes propone en (Reyes et al., 2013) un modelo de detección automática de ironía en textos basado en un set de características textuales a nivel lingüístico. Este modelo contempla dimensiones lingüisticas, sintácticas y de análisis psicológico. Los autores identificaron un set de características que permite diferenciar automáticamente un texto irónico de uno no irónico.

El modelo está organizado en un set de cuatro tipos de características conceptuales:

(a) Signos (Signatures): Este aspecto se centra en la exploración de ironía en términos de marcadores o signos específicos de textos. Se caracteriza por los elementos tipográficos como signos de puntuación y emoticonos, así como por elementos discursivos que sugieren oposición dentro de un texto. Existen tres dimensiones que representan estos signos:
- Unificación: se centra en marcas explícitas que, de acuerdo con las propiedades más relevantes de la ironía, deben reflejar una clara distinción en la información que se transmite.
- Contrafactualidad: se centra en términos discursivos que hacen alusión a la oposición o contradicción en un texto, tal como “aproximadamente”, “sin embargo” y “no obstante”.
- Comprensión temporal: se centra en la identificación de elementos relacionados con la oposición en el tiempo; es decir, términos que indican un cambio brusco en una narrativa.

(b) Lo inesperado (Unexpectedness): Se observa que la sorpresa es un componente clave en la ironía, e incluso se puede afirmar que subyace en todas las situaciones irónicas. Como consecuencia, se concibe la función “Inesperado” como un medio para capturar los desequilibrios tanto temporales como contextuales en un texto irónico. Esta función se representa en dos dimensiones:
- Desequilibrio temporal: se utiliza para reflejar el grado de oposición en un texto con respecto a la información de perfil en los tiempos presente y pasado. Por ejemplo: “Odio que cuando estoy pololeando, todas las mujeres que no me querían, ahora me busquen!”
- Desequilibrio contextual: está destinado a capturar inconsistencias dentro de un contexto. Para medir esta dimensión, se estima la similitud semántica de los conceptos de un texto en comparación con otro.

(c) Estilo (Style): El concepto de estilo se refiere a secuencias de elementos textuales que expresan características relativamente estables de cómo se aprecia un texto y que por lo tanto podría permitirnos reconocer los factores estilísticos que son indicativos recurrentes de la ironía.

(d) Escenarios emocionales (Emotional Scenarios): El lenguaje textual proporciona herramientas específicas, tales como el uso de emoticonos en redes sociales para comunicar información sobre los estados de ánimo, sentimientos y nuestros sentimientos hacia los demás. Las Expresiones irónicas a menudo usan este tipo de signos para asegurar sus efectos comunicativos (por ejemplo, "me siento tan desgraciado sin usted, que es casi como tenerlo aquí :P").

Respecto al Corpus de evaluación, se recopilaron desde Twitter un set de 40.000 tweets divididos en cuatro partes, representadas por un hashtag: #irony, #education, #humor, #politics. Por lo tanto, se asume que 10.000 tweets son irónicos y 30.000 no irónicos. Los tweets duplicados fueron removidos. Este corpus está disponible y puede ser obtenido contactando a los autores. Cabe destacar que este corpus es ampliamente utilizado por distintos trabajos relacionados a la detección de ironía.

(Charalampakis et al., 2015) proponen modelo de detección de ironía aplicado en mensajes de Twitter (tweets), los cuales están basados en un contexto de importancia para el país Grecia: en las elecciones políticas.

El modelo propuesto considera características que pretenden detectar desequilibrio y cambios inesperados en un texto, patrones que son típicos de la ironía. Cada una de estas características se reflejan en funciones, las cuales manejan parámetros que permiten contar o verificar la incidencia de cada una de estas características en un mensaje de Twitter (tweet).

A continuación se describen estas características:

(a) Comunicación mediante signos (Spoken): La ironía verbal en Twitter se expresa a menudo como conversaciones diarias entre los usuarios, utilizando en gran medida guiones (-) y asteriscos (*). Sus apariciones en los tweets incrementan positivamente un contador dedicado a esta característica. Si hay por lo menos uno de los caracteres anteriores en el tweet, el valor asignado a la función "Spoken" es "verdadero", de lo contrario es "falso".

(b) Palabras extrañas (Rarity): Los autores recopilaron un diccionario de frecuencia de 25.898 palabras el cual es analizado mediante una fórmula, la cual revisa cada palabra del tweet en el diccionario de frecuencia. Si no se encuentra la palabra allí, concede un peso de acuerdo a una escala de clasificación dentro de la función "Rarity".

(c) Significados (Meanings): Esta función utiliza la aplicación Wordnet [1], la cual tiene una utilidad llamada "BalkaNet" que sirve para extraer los significados de cada palabra. Lo anterior es aplicado considerando que el uso de una palabra con múltiples significados implica ambigüedad e ironía.

(d) Léxico (Lexical): Esta función analiza atributos léxicos en cada texto, los cuales son: letras, palabras repetidas, metáforas y la puntuación. Las letras repetidas, por ejemplo, caracterizan una expresión verbal cargada de emociones, lo cual puede llevar a expresar ironía. Por ende, cada atributo léxico incrementa un contador en la función Lexical.

(e) Uso de emoticonos (Emoticons): Esta característica detecta todas las posibles variaciones de los emoticones que representan caras sonrientes, tristes y burlescas. La existencia de emoticonos es una leve indicación de la ironía, debido a la carga emocional. De esta forma, esta función busca emoticonos en cada tweet.

[1] https://wordnet.princeton.edu/

El Corpus de evaluación fue creado por los propios autores y consistió en tweets recolectados una semana antes y una después de las elecciones parlamentarias de 2012. Una vez que se realizó la limpieza al corpus de evaluación, quedó con un total de 44.438 tweets.

(Hernandez-Farıas et al., 2015) proponen un trabajo que consiste en la definición de un set de características que combinan propiedades de textos con información desde análisis lexicón. El principal aporte de esta investigación es la utilización de la clasificación de palabras respecto a su polaridad.

La propuesta de este modelo consiste en dos características de análisis de sentimientos: puntaje de sentimientos y valor de polaridad, que miden los sentimientos involucrados en cada tweet. Estas se clasifican en características basadas en estadística y basadas en léxico.

A continuación se describen estas características:

(a) Características basadas en estadística: Son patrones textuales que pueden obtenerse tomando en cuenta la frecuencia de alguna palabra o carácter en un tweet. Se clasifican en las siguientes dimensiones:
- Marcadores textuales (función TM): Esta característica analiza la frecuencia de las señales visuales tales como: longitud del tweet, uso de mayúsculas, signos de puntuación y emoticonos.
- Contador de oposición en un texto(función CF): Se analiza la frecuencia de los términos discursivos que hacen alusión a la oposición o contradicción en un texto como por ejemplo textos que contienen el término: "sin embargo".
- Comprensión temporal (función TC): Analiza la frecuencia de los términos que identifican elementos relacionados con la oposición en el tiempo, es decir, términos que indican un cambio brusco en una narrativa.
- Palabras con características gramaticales similares (función POS): Se cuenta con un POS-tagger, desarrollado para este tipo de textos llamados ARK [2] que calculan la frecuencia de verbos, sustantivos, adjetivos y adverbios.

(b) Basados en léxico (Lexical Based): Estos patrones obtienen información a partir del análisis del contenido textual de cada tweet. Se clasifican en las siguientes dimensiones:
- Similaridad semántica o Semantic Similarity (función SIM): Consiste en la obtención del grado de inconsistencia al medir la relación entre los conceptos contenidos en cada tweet utilizando el módulo "WordNet :: Similarity".
- Valor Emocional o Emotional Value (función EV): Se calcula el valor emocional a cada tweet, teniendo en cuenta las categorías descritas por Whissel en (Whissell, 2009) en su Diccionario del afecto en Lenguaje (DAL).
- Valor de Sentimientos o Sentiment Score (función SS): Capta y analiza el sentimiento general prodominante del tweet(positivo, negativo o neutro).
- Valor de Polaridad o Polarity Value (función PV): esta función permite identificar la polaridad de un tweet, ya sea para criticar (negativo) o para alabar (positivo) algo. Se utiliza "AFINN léxico" [3], que contiene una lista de palabras marcadas con un valor de polaridad comprendida entre menos cinco (negativo) y más cinco (positivo) para

[2] http://www.cs.cmu.edu/ ark/TweetNLP/
[3] http://www2.imm.dtu.dk

cada palabra.

Este trabajo utiliza el mismo corpus de evaluación de (Reyes et al., 2013) (40000 tweets).

Por su parte, (Carvalho et al., 2009) proponen un modelo de detección de ironía en textos presentes en aplicaciones donde el contenido es generado por el usuario, es decir, redes sociales, blogs, etc. El modelo está basado en 8 patrones lingüisticos que están relacionados con la expresión de ironía en textos. Los autores destacan que algunos de estos patrones se aplican exclusivamente al idioma portugués, pero otros, son independiente del idioma, por ejemplo, el uso de emoticonos.

La mayoría de estos patrones contienen una restricción de la polaridad, representado por la expresión [4-Gram +], que requiere la presencia de al menos un adjetivo positivo o sustantivo en una ventana de cuatro palabras, mientras se excluye la aparición de cualquier elemento negativo en tales ventanas.

A continuación, se describen los patrones utilizados por este modelo.

(a) Uso de diminutivos ($P_{dim}$): Los diminutivos se utilizan comúnmente en portugués, con el propósito de expresar sentimientos positivos, tales como el afecto o la ternura. Sin embargo, también se puede utilizar con sarcasmo y la ironía para expresar un insulto o depreciación hacia la entidad que representan.

(b) Uso de formas demostrativas ($P_{dem}$): La ocurrencia de cualquier forma demostrativa, es decir, “este” (esto), “esse” y “aquele” (que), antes de una entidad humana por lo general indica que dicha entidad está siendo mencionada negativa o peyorativamente.

(c) Uso de interjecciones ($P_{itj}$): Las interjecciones abundan en los textos subjetivos, entregando valiosa información referente a los autores, tales como sus emociones, sentimientos y actitudes.

(d) Uso de pronombres ($P_{verb}$): El tipo de pronombre utilizado para referirse a las personas también puede ser una pista importante para la detección de ironía, especialmente en idiomas como el portugués, donde la elección de un pronombre o forma de expresión específica (por ejemplo, “tu” frente a “voce”, tanto traducible por” usted ”) puede depender del grado de proximidad o familiaridad entre el hablante y la entidad a la que se refiere.

(e) Uso de adjetivos valorativos ($P_{cross}$): Los adjetivos valorativos con una polaridad positiva o neutral suelen tener una interpretación negativa o irónica cada vez que aparecen en redacciones cruzadas, donde los adjetivos relacionados con el sustantivo se modifican a través de la preposición “de” (por ejemplo “ O Comunista do ministro “/” el comunista del ministro ”).

(f) Uso de signos de puntuación ($P_{punct}$): La puntuación se utiliza con frecuencia para verbalizar las emociones y los sentimientos de los usuarios y para la señalización de texto intencionalmente humorístico o irónico. Los autores suponen que la presencia en una frase de una secuencia compuesta por más de un signo de exclamación y/o de interrogación puede ser utilizado como una pista para la detección de la ironía.

(g) Uso de comillas ($P_{quote}$): Las comillas también se utilizan con frecuencia para expresar y hacer hincapié en un contenido irónico, especialmente si el contenido tiene una polaridad positiva previa (por ejemplo, un adjetivo positivo para calificar una entidad).

(h) Uso de jerga de internet ($P_{laught}$): La jerga de internet contiene una variedad de expresio-

nes generalizadas y símbolos que normalmente representan una expresión sensorial, lo que sugiere diferentes actitudes o emociones. En los experimentos, los autores han considerado: las siglas "LOL" y variaciones correspondientes (LOL), las palabras que son imitaciones lingüisticas (onomatopeyas) como "ah", "eh" y "hi" (AH) y los emoticonos positivos anteriores ":)" ";-)" y ": P".

Para generar el corpus de evaluación, los mismos autores recopilaron post de usuarios en noticias, lo que finalmente les permitió contar con un total de 1 millón de sentencias u oraciones.

(Reyes y Rosso, 2012) se centran en la identificación de los componentes claves para la detección de la ironía en textos. El modelo propuesto define características de ironía en términos de elementos lingüísticos observados en los enfoques de los trabajos anteriores al 2012. Está orientado a analizar las opiniones de los consumidores de productos de Amazon. Esta investigación inspira a (Reyes et al., 2013), trabajo que toma este estudio como base de su conjunto de características.

Se consideran 6 características que se presentan a continuación:

(a) Uso de N-gramas: Esta característica se centra en la representación de los documentos irónicos en la forma de secuencias de n-gramas (de orden 2 hasta 7) con el fin de encontrar un conjunto de palabras recurrentes que podrían expresar ironía.

(b) Uso de POS n-gramas: El objetivo de esta función es la obtención de secuencias recurrentes de patrones morfosintácticos. De acuerdo con esta definición, la ironía busca expresar un significado opuesto.

(c) Perfil "divertido": Esta característica tiene caracteriza los documentos en términos de propiedades de humor, lo anterior dado que la ironía se aprovecha de los aspectos divertidos para producir su efecto. Para representar esta función, los autores seleccionaron algunas de las mejores características relacionadas al humor que encontraron en la literatura (Mihalcea y Strapparava, 2006): características estilísticas y egocentrismo humano.

(d) Perfil "positivo y negativo": Esta característica se basa en la premisa que una de las propiedades más importantes de la ironía es la intención de comunicar información negativa a través de un información positiva.

(e) Perfil "afectivo": La característica de perfiles afectivos intenta caracterizar los documentos en términos de palabras que simbolizan contenidos subjetivos tales como las emociones, sentimientos, estados de ánimo, etc.

(f) Perfil "agradabilidad": La última característica es un intento de representar escenarios cognitivos ideales para expresar ironía. Esto significa que, así como las palabras, los contextos en los que aparece la ironía son enormes. Por lo tanto, puesto que es imposible hacer todas las combinaciones, los autores intentan definir un esquema para representar contextos irónicos favorables y desfavorables, sobre la base de los valores de amenidad.

El corpus de evaluación consta de opiniones o revisiones de clientes provistas a los investigadores por el sitio Amazon [4]. Se seleccionaron bajo los criterios de (1) Viralidad y (2) Efecto de Ironía, es decir, los opiniones de los productos más impactantes en redes sociales. La conformación del corpus de evaluación luego del filtrado quedó con 11.861 opiniones irónicas, incluídas

[4] www.amazon.com

3.000 opiniones negativas desde el mismo Amazon por cada producto.

A continuación, en la Tabla 3.3 se presenta un resumen de los distintos enfoques de detección de ironía que aplica cada trabajo estudiado. Se clasifican en: Machine Learning, Análisis Lexicón e Híbrido.

| Trabajo | Machine Learning | Análisis Lexicon | Híbrido |
|---|---|---|---|
| (Reyes et al., 2013) | | | X |
| (Reyes y Rosso, 2011, 2013; Reyes et al., 2012) | | | X |
| (Charalampakis et al., 2015) | X | | |
| (Hernandez-Farıas et al., 2015) | | | X |
| (Carvalho et al., 2009) | | | X |
| (Reyes y Rosso, 2012) | | | X |

Tabla 3.3: Resumen de enfoques de análisis de sentimiento.

### 3.3.1. Rendimientos de los Modelos de Detección de Ironía

A continuación se presentan en la Tabla 3.4 los distintos resultados de rendimiento de cada modelo estudiado. Se consideran 3 métricas de clasificación (Olson y Delen, 2008):

- **Precision:** Corresponde al ratio entre el número de instancias relevantes recuperadas y el número de instancias recuperadas. Mientras el valor de Precision se acerque más a 1, mayor porcentaje de instancias recuperadas son consideradas relevantes y si se acerca a 0, menor porcentaje de instancias recuperadas no son consideradas relevantes.
- **Recall:** Expresa la proporción entre el número de instancias relevantes recuperadas y el total de instancias recuperadas relevantes totales. Mientras el valor de Recall se acerque más a 1, mayor porcentaje de instancias recuperadas son consideradas relevantes y si se acerca a 0, menor porcentaje de instancias recuperadas no son consideradas relevantes.
- **F-Measure:** Se considera la media armónica que combina los valores de las métricas Precision y Recall. Si el valor de F-Measure se acerque a 1 indica que se está dando la misma ponderación a Precision y Recall. Si es mayor que 1, indica que se da mayor ponderación a Recall y si se acerca a 0, mayor ponderación a Precision.

## 3.4. Modelos de Detección de Sarcasmo

Los trabajos revisados que abordan la detección de sarcasmo en texto utilizan técnicas muy similares a las utilizadas en la detección de ironía. A continuación se describen dichos trabajos.

El trabajo de (Lunando y Purwarianti, 2013) propone un modelo de detección de sarcasmo basado en características típicas para el idioma Indoneso. Dichas características son:

(a) Unigramas: Los autores analizaron los unigramas más relevantes del idioma indonesio, los cuales son:

| Trabajo | Rendimiento |
|---|---|
| (Reyes et al., 2013) | 75 % en métrica Precision |
| (Reyes y Rosso, 2011, 2013; Reyes et al., 2012) | entre 72 y 82 % en métrica Precision |
| (Charalampakis et al., 2015) | 82.4 % en métrica Precision |
| (Hernandez-Farıas et al., 2015) | entre 70 y 80 % métrica F-Measure |
| (Carvalho et al., 2009) | entre 45 y 85 % en métrica Precision |
| (Reyes y Rosso, 2012) | entre 72 y 89 % en métrica Precision |

Tabla 3.4: Resumen de rendimientos por modelos estudiados.

- Negación: Cuando una palabra positiva va precedida por una de negación, se convierte en una negativa.
- Contexto de la palabra: Dependiendo el contexto, una palabra puede cambiar su sentimiento, es decir, una palabra neutra podría transformarse en una que exprese sentimiento.
- Afijos: Una misma palabra, con distintos afijos, puede tener distintos sentimientos.

(b) Negatividad: Esta característica representa el porcentaje global del sentimiento negativo en el tema de un texto. Proporciona información sobre el sentimiento real de un tema determinado.

(c) Número de palabras que contienen interjecciones: Esta función muestra el número de interjecciones [5] en las palabras. Los autores emplearon esta característica basada en la observación de que entre 100 texto sarcasmo, hay 20 que tienen interjecciones.

(d) Preguntas: Esta característica cuenta el número de palabras asociadas a preguntas (qué, cómo, cuándo, donde, etc) en un texto.

Por otra parte, en (Nagwanshi y Veni Madhavan, 2014) se plantea un modelo de detección de sarcasmo basado en Machine Learning que tiene dos módulos: La extracción de Características y Clasificación automática de sarcasmo. Además, al igual que la mayoría de los trabajos revisados, también se identificaron un set de características típicas en la detección de sarcasmo, las cuales se presentan a continuación:

(a) Léxicas: se basa en n-gramas que se producen más de dos veces en los datos de entrenamiento, con los cuales los autores construyeron un diccionario de estas palabras.

(b) Sintácticas: es la combinación de etiquetas de part-of-speech. Los autores determinaron en sus estudios que los adverbios se utilizan a menudo en expresiones sarcásticas. La presencia de adjetivos parecer ser una característica discriminativa.

(c) Semánticas: determina si existen palabras que se contradicen. Por ejemplo, "Me encanta ser ignorado." En el ejemplo se observan dos palabras casi opuestas que se utilizan en una expresión sarcástica.

(d) Pragmática: determina el momento en el sentimiento de la oración difiere de los emoticonos.

[5] Interjección, según RAE: Clase de palabras invariables, con cuyos elementos se forman enunciados exclamativos, que manifiestan impresiones, verbalizan sentimientos o realizan actos de habla apelativos.

Por ejemplo: "Me encanta despertar temprano en la mañana :@", seguido por el emoticono de ceño fruncido. En el ejemplo, el sentimiento de la frase se supone que es positivo, pero el uso del emoticono ceño fruncido hace que sea sarcástico.

(e) Cortesía: determinará si hay palabras como "extremadamente", "demasiado" que se utiliza antes de cualquier palabra que lleva el sentimiento positivo. Esta característica es muy útil en la deteción de sarcasmo porque cuando sarcásticamente queremos expresar algo, entonces decimos: "Usted es extremadamente bueno", en lugar de declaración como "usted es bueno".

(f) Cambio de Sentimientos: se refiere a la diferencia entre sentimientos en una oración. Por ejemplo, "Me encanta cuando estoy enfermo y ni siquiera soy capaz de dormir." En este ejemplo el sentimiento de ambas cláusulas es diferente, por un lado alegría y por otro tristeza.

## 3.5. Comparación entre Modelos de Detección de Ironía y Sarcasmo

Existe bastante similitud en el planteamiento que tienen los modelos de ironía y sarcasmo revisados en este trabajo. Todos proponen sets de características que son propias de la ironía y el sarcasmo, las cuales ayudan a caracterizar un mensaje escrito y analizarlo en función de cada una de estas características. Si el texto es representativo, es decir, contiene una o más de las características definidas, se trata de ironía/sarcasmo, de lo contrario, es una opinión objetiva. En ese sentido, todos los trabajos revisados acerca de detección de sarcasmo identifican casi las mismas características que los modelos de detección de ironía, lo que hace presumir que la definición de ironía y sarcasmo utilizada en los trabajos revisados, apuntan prácticamente a lo mismo, es decir, expresión de una opinión contraria a la que se escribe en el texto.

A continuación se analizan las características más comunes encontradas en los trabajos revisados:

(a) **Uso de Marcadores Textuales:** Esta característica se refiere a la presencia de signos de puntuación, símbolos de pregunta, exclamación, uso de mayúscula, etc. en mensajes de texto. Esta característica es abordada por (Reyes et al., 2013) ,(Charalampakis et al., 2015), (Hernandez-Farıas et al., 2015) y (Carvalho et al., 2009)

(b) **Uso de Emoticonos:** El uso de emoticonos en un mensaje de texto (caras felices, tristes, sacando lengua, enojado, risa, etc) presume sentimientos que pueden conducir a ironía/sarcasmo en el mensaje. Esta característica es incluída en los trabajos de (Reyes et al., 2013), (Charalampakis et al., 2015), (Hernandez-Farıas et al., 2015), (Carvalho et al., 2009) y (Reyes y Rosso, 2012)

(c) **Análisis gramatical de las palabras:** La mayoría de los trabajos realiza un análisis gramático de palabras en función de sus características sintáxicas y semánticas para buscar características de ironía utilizando POS, ngramas y skipgramas. Los autores (Reyes et al., 2013), (Charalampakis et al., 2015), (Hernandez-Farıas et al., 2015), (Carvalho et al., 2009) , (Reyes y Rosso, 2012), (Lunando y Purwarianti, 2013) y (Nagwanshi y Veni Madhavan, 2014) utilizan en sus modelos este tipo de análisis.

(d) **Análisis de oposición y negación del mensaje:** Los autores (Reyes et al., 2013), (Hernandez-Farıas et al., 2015), (Reyes y Rosso, 2012) y (Nagwanshi y Veni Madhavan, 2014) mediante análisis gramátical y la búsqueda de patrones linguísticos buscan identificar mensajes con oposición de opinión y cambio drástico de sentimientos (negación) en el mensaje.

En las tablas 3.5 y 3.6 se resume la comparación entre todos los trabajos revisados.

| Trabajo | Año | Fuente | Tamaño(tweets/oraciones) | Idioma |
|---|---|---|---|---|
| (Reyes et al., 2013) | 2013 | Elaboración propia | 40000 tweets | Inglés |
| (Charalampakis et al., 2015) | 2015 | Elaboración propia | 44438 tweets | Griego |
| (Hernandez-Farıas et al., 2015) | 2015 | Mismo que (Reyes et al., 2013) | 40000 tweets | Inglés |
| (Carvalho et al., 2009) | 2009 | Elaboración propia | 1 millón de oraciones | Portugués |
| (Reyes y Rosso, 2012) | 2012 | Elaboración propia | 11861 opiniones de usuario | Inglés |
| (Lunando y Purwarianti, 2013) | 2013 | Elaboración propia | 980 tweets | Indonesio |
| (Nagwanshi y Veni Madhavan, 2014) | 2014 | Elaboración propia | 400 tweets | Inglés |

Tabla 3.5: Resumen respecto a los corpus de evaluación.

## 3.6. Conclusión del Capítulo

Mediante una revisión sistemática de literatura se pudieron revisar los principales conceptos relacionados con la detección de ironía en textos. Se estudiaron los principales trabajos relacionados con la detección de ironía y sarcasmo y cómo estos pueden complementarse entre si. Se analizaron sus rendimientos y características comunes, las cuales son la base para el nuevo modelo de detección de ironía que se propone en este trabajo. La Tabla 3.6 resume las características comunes de todos los trabajos revisados.

| Trabajo | Marcadores Textuales | Emoticonos | Análisis gramatical | Oposición y negación | Aplicable Español? |
|---|---|---|---|---|---|
| (Reyes et al., 2013) | Si | Si | Si | Si | Si |
| (Charalampakis et al., 2015) | Si | Si | Si | No | Si |
| (Hernandez-Farıas et al., 2015) | Si | Si | Si | Si | Si |
| (Carvalho et al., 2009) | Si | Si | Si | No | Si |
| (Reyes y Rosso, 2012) | No | Si | Si | Si | Si |
| (Lunando y Purwarianti, 2013) | No | No | Si | No | Si |
| (Nagwanshi y Veni Madhavan, 2014) | No | No | Si | Si | Si |

Tabla 3.6: Resumen respecto a características comunes de los modelos revisados.

En el próximo capítulo, se revisará en detalle el proceso de construcción del corpus de evaluación para el nuevo modelo de detección de ironía.

# Capítulo 4

# Diseño y Construcción del Corpus de Evaluación

El objetivo principal de este trabajo es la propuesta de un modelo de detección de ironía y para evaluarlo se construyó un corpus de evaluación constituído por textos, específicamente, mensajes de Twitter (tweets). El diseño y la construcción de este corpus esta basada en el trabajo de (Bosco et al., 2015), el cual describe el proceso de desarrollo de un corpus para el idioma italiano, compuesto de 3288 tweets que fueron recopilados usando temáticas relevantes de Italia. Esto tweets fueron obtenidos directamente desde la página de Twitter y luego clasificados en irónicos, positivos, negativos, mixtos y neutros por 5 personas, las cuales dividieron el corpus en conjuntos de 200 tweets logrando un acuerdo general sobre las etiquetas.

A continuación, se decriben los pasos y características de nuestro corpus construído.

## 4.1. Dominio de la temática de las opiniones

Con la finalidad de obtener una cantidad similar de opiniones positivas, negativas y casos de ironía, se seleccionaron temáticas relacionadas con la política Chilena predominante en el año 2016. Estos temas se escogieron debido a que fueron tendencia nacional [1] durante el año 2016 y abarcan dominios que en Chile son controversiales, por ejemplo, política, educación, igualdad de género y demandas sociales. Se incluyeron 7 temáticas, las cuales distribuyen los dominios descritos anteriormente. Estas temáticas son:

- Nueva Mayoría
- Sebastián Piñera
- AFP
- Ni una menos
- Camila Vallejo
- Trump President
- Ley de etiquetado

[1] https://twitter.com/trendingtopics

## 4.2. Obtención de los Tweets

Inicialmente, se utilizó la API de Twitter para Java y PHP, para poder parametrizar las consultas y obtener masivamente datos en formato JSON, y luego almacenarlos en una base de datos MySQL. La API Search de Twitter presentó el problema que solo permite ventanas de consultas cada 15 minutos, lo cual dificultó el proceso. Para solucionar este problema, se desarrolló una pequeña herramienta en Javascript que permite analizar la estructura del documento de la página Twitter (su árbol DOM [2]) del sitio Twitter y así utilizar directamente la búsqueda desde el sitio web, y luego cada uno de los resultados se serializa a formato JSON y se envía en conjunto a un servidor donde son almacenados en la base de datos. Finalmente, se obtuvieron un total de 13.500 tweets de todas las temáticas descritas anteriormente.

## 4.3. Filtrado Manual

La desventaja de utilizar la herramienta escrita en Javascript descrita anteriormente fue que sólo se podía obtener el mensaje del tweet, y no saber si es un retweet, una conversación, si es publicidad, etc. Por lo tanto, se realizó un procesamiento manual, el cual consistió en revisar los tweets obtenidos y declararlos aptos o no aptos para formar parte del corpus. Los criterios utilizados fueron:

- No ser un retweet
- No ser publicidad
- No ser el encabezado de una noticia o nota
- No promocionar sitios web ni empresas
- Descartar tweets de conversaciones, ya que no existe forma de conocer el contexto de la conversación completa

Finalmente, luego de esta etapa de filtrado, se obtuvieron 3.021 tweets que pueden formar parte de la encuesta de etiquetado.

## 4.4. Preparación de la Encuesta

Para el etiquetado del corpus se desarrolló una encuesta a través de un sistema web desarrollado para este fin, que permite dividir la base de datos de tweets en porciones para asignarlas a revisores. Cada revisor debe etiquetar una cantidad de tweets utilizando los siguientes criterios descritos en la encuesta:

- **Irónico:** El tweet representa un mensaje irónico, ya que el emisor quizo expresar lo contrario a lo que escribió.

La encuesta se realizó mediante un sistema web que exhibia el tweet y daba la opción a clasificar preguntando “Irónico?”, donde marcar el checkbox significaba que el tweet era irónico.

[2] https://www.w3.org/2005/03/DOM3Core-es/introduccion.html

## 4.5. Resultados

El grupo de evaluación se conformó por 7 personas, las que se pertenecen al rango etario entre los 20 y 30 años y con niveles educacionales de profesionales titulados y un estudiante. No se exigió ninguna habilidad o criterio especial para la selección de evaluadores. 6 evaluadores se dedicaron a una porción de 500 tweets aproximadamente, los cuales corresponden a porciones distintas de un total de 3.021 tweets. Un evaluador tomó la evaluación del conjunto completo, es decir 3.021 tweets en total. En definitiva, cada tweet fue evaluado por más de una persona. La concordancia entre las respuestas de los evaluadores se analizó utilizando la medida **Kappa de Cohen** que permite realizar un análisis descriptivo, obteniendo el grado de concordancia que tienen un par de evaluadores al etiquetar sujetos según un conjunto de clases (J. L. Fleiss y Paik, 2013). Nos basamos en la medida utilizada en (Bosco et al., 2015) que establece un valor $k = 0{,}65$, el cual indica que se acepta aproximadamente un 65 % de concordancia para considerar un acuerdo entre evaluadores.

Las discrepancias (cerca del 30 % de los casos) fueron tratadas mediante una segunda opinión en particular sólo cada caso, y en los casos en que persistía la discrepancia, se descartaron.

Finalmente, se obtuvieron un total de 555 tweet irónicos, los cuales se detallan en la Tabla 4.1 y 2.200 tweets no irónicos. Fueron descartados un total de 266 tweets, debido a que no se logró una concordancia en la segunda instancia de evaluación descrita anteriormente.

| Resultados | |
|---|---|
| Irónicos | 555 |
| No Irónicos | 2.200 |
| Descartados | 266 |

Tabla 4.1: Resultados Etiquetado del Corpus.

Donde el 80 % aproximadamente corresponden a tweets no irónicos y el 20 % a tweets irónicos, tal como se puede apreciar en la Figura 4.1.

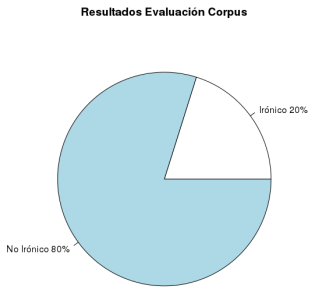


Figura 4.1: Porcentajes de tweets irónicos y no irónicos del corpus.

El corpus está compuesto de **49.442** palabras, y las más frecuentes se muestran en la Tabla

4.2 y en la Figura 4.2

| Palabras más Frecuentes | |
|---|---|
| de | 2.043 |
| la | 1.605 |
| que | 1.491 |
| a | 1.410 |
| y | 1.291 |
| el | 1.226 |
| no | 1.126 |
| en | 853 |
| es | 716 |
| los | 653 |
| por | 618 |
| con | 591 |
| se | 533 |
| para | 454 |
| su | 439 |
| lo | 432 |
| q | 427 |
| un | 412 |
| las | 372 |
| camila | 371 |
| vallejo | 343 |
| piñera | 337 |
| le | 328 |
| del | 320 |
| al | 285 |
| si | 283 |

Tabla 4.2: Palabras más frecuentes en el Corpus.

Además, los emoticonos encontrados que son más frecuentes se muestran en la Tabla 4.3

## 4.6. Conclusión del Capítulo

Se construyó un corpus de evaluación compuesto de mensajes de Twitter (tweets) los cuales fueron recopilados usando aplicaciones que permiten almacenarlos en una base de datos. Estos tweets fueron escogidos respecto a temáticas relevantes para Chile el año 2016. Participaron 7 evaluadores, los cuales evaluaron 3.021 tweets, de los cuales, 266 fueron descartados, 2.200 fueron etiqutados como tweets no irónicos y 555 como tweets irónicos. El corpus quedó compuesto por


Figura 4.2: Nube de Palabras del Corpus.

| Emoticonos más Frecuentes | |
|---|---|
| :/ | 45 |
| xD | 30 |
| :( | 7 |
| :D | 6 |
| <3 | 5 |
| :) | 4 |
| ;) | 4 |
| (8 | 2 |
| ): | 2 |
| XD | 2 |
| xd | 1 |
| :-) | 1 |
| :P | 1 |
| =P | 1 |

Tabla 4.3: Emoticonos más frecuentes en el Corpus.

2.755 elementos, con un porcentaje de 80 % no irónicos y 20 % irónicos. A continuación, se decribe el nuevo modelo de detección de ironía, el cual es evaluado utilizando el corpus creado en este capítulo.

# Capítulo 5

# Modelo de Detección de Ironía en Español

Esta propuesta de modelo está basada en las caractarísticas más utilizadas por los modelos de detección de ironía estudiados en este trabajo. Como se ha descrito anteriormente, existen características comunes entre todos los modelos revisados, las cuales, serán tomadas como base de este nuevo modelo, debido a que han sido probadas para distintos idiomas y corpus de evaluación. Además, se proponen dos nuevas características no encontradas en los modelos revisados.

La Figura 5.1 describe el modelo, el cual consta de diversas etapas que comienzan, dependiendo la característica, con el preprocesado del tweet para cada elemento del corpus y luego con el procesado de cada característica que compone el modelo. Esto permite preparar la entrada para la clasificación automática que se realiza mediante la aplicación **Weka** [1]. Una vez que han sido generadas las entradas, se evalúan las distintas combinaciones de atributos con las cuales se obtengan los mejores rendimientos. Lo anterior, dependerá del corpus de evaluación, ya que dependiendo del dominio temático de la selección de tweets, podrán variar la frecuencia de aparición de los elementos típicos de ironía que conforman este nuevo modelo.

Este modelo de detección de ironía está compuesto de las siguientes características:

(a) Uso de emoticonos
(b) Uso signos de puntuación
(c) Uso de palabras en mayúscula
(d) Presencia de signos típicos de ironía
(e) Uso de adverbios no temporales
(f) Presencia de contradicción texto-emoticón
(g) Uso de adverbios temporales
(h) Análisis de ngramas
(i) Análisis de skipngramas

Algunas de estas características requieren pasar por la etapa de **pre-procesado**. El preprocesado se encarga de preparar cada elemento del corpus en función de las necesidades de cada

[1] http://www.cs.waikato.ac.nz/ml/weka/

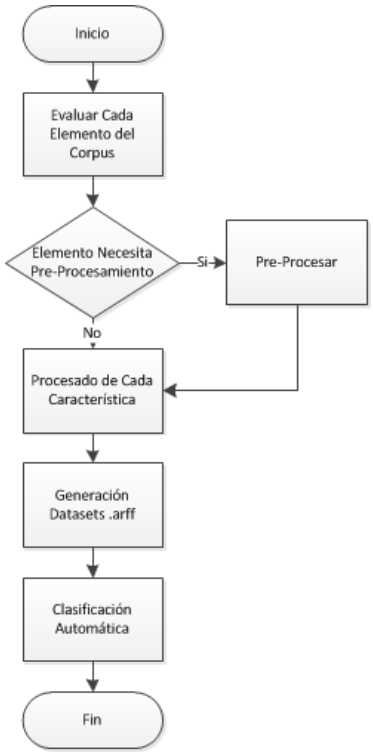


Figura 5.1: Flujo del modelo de detección de ironía.

característica del modelo de detección de ironía en textos. Cada una de sus funciones se aplican dependiendo de cada característica a cada elemento del corpus y no siempre siguen la misma secuencia. Es decir, que siempre va a depender de cada característica del modelo, las funciones de pre-procesamiento necesarias y su respectivo órden de aplicación.

A continuación, se detallan cada una de las características de este modelo de detección de ironía, además de su procesamiento y cálculo.

## 5.1. Características del Modelo

### 5.1.1. Uso de Emoticonos

Los emoticonos generalmente están presente en textos irónicos (Reyes et al., 2013), por lo tanto, esta característica busca emoticonos en cada tweet a analizar. Para ello, se cuenta con un diccionario con 300 emoticonos de uso internacional de elaboración propia. Por ejemplo:

**Ejemplo 5.1**
- ***:O*** *mi jefe otra vez llegando temprano* ***xD***
- *Por fin llegó mi encargo de China, lloraré ahora* ***:')***

Los elementos marcados en negrita corresponden a emoticonos que se buscan en cada tweet a analizar utilizando el diccionario elaborado. De esta forma, la aparición de uno o más emoticonos

incrementará un contador, el cual, cuán mayor sea, mayor probabilidad existirá que el tweet analizado sea irónico. Esto implica que esta característica se calcula respecto a la siguiente Fórmula:

$$\text{Uso de Emoticonos} = \sum_{i=0}^{n} Emoticono \tag{5.1}$$

Donde **Emoticono** es igual a 1 cuando se encuentra un emoticono del diccionario dentro de un tweet. Lo anterior, implica entonces sumar todos los emoticonos presentes en un tweet.

Para procesar esta característica se pre-procesa cada tweet identificando los emoticonos que contenga. Para esto y como se mencionó anteriormente, se tiene un diccionario de elaboración propia conformado por la mayoría de los emoticonos utilizados en redes sociales. En el Ejemplo 5.1 la sumatoria **Uso de Emoticonos** descrita en la Ecuación 5.1 entregaría un valor igual a 2, ya que el texto contiene 2 emoticones (:O y xD).

### 5.1.2. Uso Signos de Puntuación

En los textos donde se expresa ironía generalmente se utilizan uno o más signos de puntación (Reyes et al., 2013) (Hernandez-Farıas et al., 2015) (Carvalho et al., 2009). Para analizar esta característica, el tweet pasa por un pre-procesamiento en el cual se utiliza una expresión regular para encontrar signos de puntuación en cada tweet. Por ejemplo:

**Ejemplo 5.2** ***Oh por Dios!!!*** *el gobierno se hizo cargo de un problema****?! ...*** *sorprendido****!***

Los elementos marcados en negrita corresponden a signos de puntuación que se buscan en cada tweet a analizar utilizando la expresión regular adecuada. Así, la aparición de signos de puntuación en un tweet incrementará un contador, el cual, mientras mayor sea, mayor probabilidad existe que el tweet sea considerado irónico. Lo anterior, se calcula mediante la Fórmula:

$$\text{Uso de Signos de Puntuación} = \sum_{i=0}^{n} Signo \tag{5.2}$$

Donde **Signo** es igual a 1 cuando se encuentra un signo de puntuación dentro de un tweet. Lo primero a realizar en este procesamiento es el pre-procesado, que consiste en tokenizar el tweet mediante una expresión regular que detecta los signos de puntuación en cada token. Esta tokenización solo entregará valores correspondientes a signos de puntuación, los cuales son contados por cada tweet. En el Ejemplo 5.2 el tweet se descompondría en tokens de la forma [!,!,!,!,!,?,.,.,.] y entregaría un valor de **Signo** igual a 9.

### 5.1.3. Uso de Palabras en Mayúscula

Es también frecuente encontrar en los textos que expresan ironía el uso de palabras en mayúscula (Reyes et al., 2013). Para obtener información de esta característica, se utilizan funciones que permiten verificar si una palabra está o no escrita usando mayúsculas. Por ejemplo:

**Ejemplo 5.3** ***NO LO CREO*** *mi amigo me pagará este viernes* ***LLOVERÁ****!*

De esta forma, se busca contar la cantidad de palabras escritas en mayúscula en cada tweet, de acuerdo a la siguiente Fórmula:

$$\text{Uso de Palabras en Mayúscula} = \sum_{i=0}^{n} \text{Mayúscula} \tag{5.3}$$

Donde **Mayúscula** es igual a 1 cuando se encuentra una palabra escrita en mayúscula dentro de un tweet. Mientras mayor sea el número de palabras escritas en mayúscula, mayor probabilidad existirá que el tweet sea irónico. Para procesar el uso de palabras en mayúscula, se tokeniza cada tweet y se verifica si la palabra está compuesta por letras mayúsculas en su totalidad. Si esto ocurre, se cuenta como 1 palabra mayúscula. En el Ejemplo 5.3 la Fórmula 5.3 entregaría un valor de palabras mayúsculas igual a 4.

### 5.1.4. Signos típicos de ironía

Esta característica es original de este trabajo. Existen distintos sígnos, frases y emoticonos que indican presencia de ironía por si solos. A diferencia del punto anterior en que se busca la presencia de algunos emoticonos, esta característica busca específicamente un conjunto de elementos obtenidos de un diccionario de elaboración propia que contiene palabras, emoticonos y elementos típicos de textos irónicos. Por ejemplo:

**Ejemplo 5.4** *Estoy feliz, mi marido está trabajando* ***(?)***

En el ejemplo anterior, el elemento **(?)** es ampliamente utilizado para expresar que un texto es irónico. Para calcular esta característica, se deben sumar todos los signos típicos de ironía presentes en un tweet, de acuerdo a la siguiente Fórmula:

$$\text{Signos Típicos de Ironía} = \sum_{i=0}^{n} \text{Signo} \tag{5.4}$$

Donde **Signo** es igual a 1 cuando se encuentra un signo típico de ironía dentro de un tweet. Mientras mayor sea el número de signos típicos de ironía, mayor probabilidad existirá que el tweet sea irónico. El procesamiento de los signos típicos de ironía consiste en comparar cada token de un tweet con el diccionario de signos típicos de ironía. La ocurrencia de uno o más de un elemento en un tweet incrementa en 1 el valor de signos típicos de ironía. El Ejemplo 5.4 entregaría un valor de signos típicos igual a 1.

### 5.1.5. Uso de adverbios no temporales

Para analizar esta característica se analizan los textos en busca de adverbios que indiquen oposición o negación en la narrativa. Se cuenta con un diccionario de adverbios adaptados al español utilizados por (Reyes et al., 2013) los cuales son buscados en cada texto a analizar. Este listado se encuentra disponible en los anexos de este trabajo. Por ejemplo:

**Ejemplo 5.5** *Te quiero en mi vida* ***pero*** *de lejos!*

Para calcular esta característica suma la cantidad de veces que aparecen adverbios no temporales en un tweet, de acuerdo a la siguiente Fórmula:

$$\text{Adverbios No Temporales} = \sum_{i=0}^{n} \text{Adverbio} \tag{5.5}$$

Donde **Adverbio** es igual a 1 cuando se encuentra un adverbio no temporal dentro de un tweet. Mientras mayor sea el adverbios no temporales, mayor probabilidad existirá que el tweet sea irónico. Para el procesado de esta característica se tienen dos diccionarios de adverbios típicos de ironía utilizados por (Reyes et al., 2013) divididos en adverbios temporales y no temporales. De cada tweet se obtiene su tag usando un POS-Tagger y cada adverbio es comparado con los diccionarios. En este caso, se compara con el diccionario de adverbios no temporales. El Ejemplo 5.5 la Fórmula 5.5 entregaría un valor de adverbios temporales igual a 2.

### 5.1.6. Contradicción Texto-Emoticón

Esta característica consiste en separar el texto de los emoticones que puedan contener. (Nagwanshi y Veni Madhavan, 2014) realiza una aproximación a este tipo de análisis. Naturalmente, quedan fuera de análisis todos los textos que no contengan emoticonos. Primero se analiza la polaridad del texto utilizando análisis lexicón, dado que se cuenta con un lexicón etiquetado con palabras en español y "chilenismos" (palabras típicas de Chile) y se determina la polaridad del texto. Luego, se analizan los emoticonos, también con un lexicón etiquetado. Si existe una diferencia o contradicción entre la polaridad del texto y la del/los emoticonos, se puede tratar de un texto irónico. Por ejemplo:

**Ejemplo 5.6** *Que pena más terrible. Hoy no hay clases. :D :D :D*

Para realizar este procesamiento lo primero que ocurre es la determinación de la polaridad de cada palabra del tweet, por lo cual, se tokeniza y consulta la polaridad con 3 lexicones disponibles: fullStrenghtLexicon (Perez-Rosas et al., 2012), ML-Lexicon [2] y Chilenismos (Koza et al., 2015). Una vez se obtiene la polaridad de cada palabra, se procede a analizar el texto en busca de emoticonos y si existen, se determina su polaridad definida por un pequeño lexicón de elaboración propia. Si la polaridad general, determinada por la polaridad predominante en el texto es distinta a la del o los emoticonos, se produce la situación de contradicción. Para el texto del Ejemplo 5.6 predomina la polaridad negativa, es decir, [Que = NEU, pena = NEG, más = NEU, terrible = NEG, Hoy = NEU, no = NEG, hay = NEU, clases = NEU] muestra claramente la predominancia de la polaridad negativa y los emoticonos [:D = POS, :D = POS, :D = POS] está marcado por la polaridad positiva. En este caso, como existe contradicción, el valor de contradicción es 1. Caso contrario sería 0.

[2] http://timm.ujaen.es/recursos/ml-senticon

### 5.1.7. Uso de adverbios temporales

En este análisis, se siguen prácticamente los mismos pasos que en el análisis no temporal de adverbios que busca analizar cambios abruptos en la narrativa del texto (Reyes et al., 2013), con la diferencia que el diccionario de adverbios utilizado en esta característica contiene específicamente aquellos que indican contradicciones a lo largo del tiempo. Este diccionario es una adaptación al propuesto por (Reyes et al., 2013) y el listado se encuentra disponible en los anexos de este trabajo. Por ejemplo:

**Ejemplo 5.7** ***Hoy*** *puro estudio,* ***mañana*** *fiesta hasta perder la consciencia!*

Al igual que en la característica de adverbios no temporales, se suma la cantidad de veces que aparecen adverbios no temporales en un tweet, de acuerdo a la siguiente Fórmula:

$$\text{Adverbios Temporales} = \sum_{i=0}^{n} \text{Adverbio} \tag{5.6}$$

Donde **Adverbio** es igual a 1 cuando se encuentra un adverbio temporal dentro de un tweet. Mientras mayor sea el adverbios temporales, mayor probabilidad existirá que el tweet sea irónico. Para el procesado de esta característica se tienen dos diccionarios de adverbios típicos de ironía utilizados por (Reyes et al., 2013) divididos en adverbios temporales y no temporales. De cada tweet se obtiene su tag usando un POS-Tagger y cada adverbio es comparado con los diccionarios. En este caso, se compara con el diccionario de adverbios temporales. El Ejemplo 5.7 la Fórmula 5.6 entregaría un valor de adverbios temporales igual a 2.

### 5.1.8. Uso de ngramas

Un n-grama es una cadena de n caracteres extraída de un texto, las cuales se relacionan con la efectividad para caracterizar los estilos de escritura de un autor (Kešelj et al.). Para realizar este análisis se consideran n-gramas de entre 3 y 5 caracteres, tal como en (Reyes et al., 2013). Estas cadenas de caracteres se buscan en textos que representan ironía, con la finalidad de conocer cuales de ellas son las que están más presente en este tipo de textos. Luego, la frecuencia de estas es comparada en el texto a evaluar.

Por ejemplo:

**Ejemplo 5.8** *Qué bonito perro*

- ***3-grama****: que, ue_, e_b, _bo, bon, oni, nit, ito, to_, o_p, _pe, per,err, rro*
- ***4-grama****: que_, ue_b, e_bo, boni, onit, nito, ito_, to_p, o_pe, _per, perr, erro.*

En el ejemplo anterior, se observan todos los 3-ngramas generados para la cadena “Qué bonito perro”, es decir todas las subcadenas de 3 caracteres que se puedan formar a lo largo de la cadena principal. El mismo caso para los 4-gramas, donde las subcadenas están formadas por 4 caracteres.

Para el procesamiento del uso de ngramas se genera un vocabulario de ngramas (entre 3 y 5 gramas) utilizando un set de entrenamiento de tweets etiquetados como irónicos. Luego, para

cada tweet del corpus se genera un vector que representa el modelo Bag of Words (Tsai y Chih-Fong, 2012) que representa la frecuencia de aparición de cada ngrama del vocabulario generado en el tweet a evaluar. Debido a que el tamaño del vocabulario de ngramas generado es del órden de los 38 mil elementos, se plantea un ejemplo sencillo. De esta forma, suponiendo que el set de entrenamiento estuviera compuesto por los siguientes textos:

- T1:"Tengo dos amigos en mi pueblo"
- T2:"Tengo compadres amigos en mi casa"

El Bag of Word estaría compuesto de la siguiente forma:

| | tengo | dos | amigos | compadres | en | mi | pueblo | casa |
|---|---|---|---|---|---|---|---|---|
| T1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| T2 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

Tabla 5.1: Ejemplo funcionamiento del modelo Bag of Words.

Se puede observar en este ejemplo, que el vocabulario lo conforman las palabras [tengo , dos , amigos , compadres , en , mi , pueblo , casa] y cada Bag of Words de los textos **T1** y **T2** es un vector compuesto por la aparición en cada palabra del vocabulario en el texto. En el caso de nuestro análisis de uso de ngramas, el vocabulario está compuesto por ngramas y los valores del Bag of Words se obtienen calculado la frecuecia de aparición de cada elemento del vocabulario en el tweet a evaluar.

### 5.1.9. Uso de skipngramas

Los skipngramas son cadenas de palabras que forman parte de un texto incluyendo las cadenas que se forman entre "saltos" de palabras, es decir, omitiendo palabras que se encuentren entre ellas. Esta característica, tal como en (Reyes et al., 2013), utiliza palabras bigramas (compuestas de dos palabras) que consideren saltos de 3 palabras, incluyendo cadenas formadas por saltos de dos, una y ninguna palabra. Por ejemplo:

**Ejemplo 5.9** *Qué hermoso día para ir a pescar*

- ***3-skip-2-grama**: que hermoso, que día, que para, hermoso día, hermoso para, hermoso ir, día para, día ir, día pescar, para ir, para a, para pescar, ir a, ir pescar.*

En el ejemplo se pueden observar todos los skipngramas formados del texto "Qué hermoso día para ir a pescar", los cuales corresponden a 2-gramas formados a partir de "saltos" de hasta 3 posiciones dentro de la cadena de texto, es decir, cadenas formadas por 2 palabras que se forman al saltar 1, 2 y 3 posiciones dentro de la cadena principal. Para procesar los skipgramas se realiza un procedimiento prácticamente idéntico al descrito anteriormente con los ngramas, con la diferencia que el pre-procesado incluye transformar todas las palabras a minúsculas y eliminar todo caracter y token que no sea alfanumérico. De esta forma, a diferencia de obtener ngramas acá se obtienen palabras y "saltos" entre ellas. Se construye un vocabulario de skipngramas y se

genera un Bag of Word de la frecuencia de cada elemento del vocabulario para cada tweet del corpus.

## 5.2. Clasificación Automática

Este modelo sigue el enfoque de machine learning llamado **aprendizaje supervisado**, el cual consiste en que un supervisor entregue un set de entrenamiento etiquetado con los valores correctos que se deberían obtener, dado un input (Dietterich et al., 2012). Estos datos de entrenamiento, tienen etiquetas determinadas que hacen referencia a una clase en particular. Además de estas etiquetas, se necesitan atributos característicos de cada clase que permitan realizar predicciones sobre nuevos datos de input. Por ejemplo, en el caso de este trabajo, se busca clasificar en clases **irónicas** y **no irónicas** y los atributos corresponden a las características que conforman este modelo de detección de ironía.

### 5.2.1. Clasificador Naive Bayes

Para este experimento se utilizó como clasificador el algoritmo **Naive Bayes**, debido a que ampliamente utilizado en los modelos de detección de ironía revisados en este trabajo: (Reyes et al., 2013), (Charalampakis et al., 2015), (Reyes y Rosso, 2012), (Wang et al., 2015), (Lunando y Purwarianti, 2013) y (Hernandez-Farıas et al., 2015). El clasificador Naive Bayes consiste en la predicción de la probabilidad de que un elemento pertenezca a cierta clase basandose en el teorema de Bayes (Jiawei Han y Pei, 2011)

$$P(c_i|X) = \frac{P(X|c_i)P(c_i)}{P(X)} \tag{5.7}$$

La Ecuación 5.7 muestra los fundamentos del algoritmo Naive Bayes, donde dado un vector de atributos X, se obtiene la probabilidad de que pertenezca a una clase $c_i$.

### 5.2.2. Configuración

Para realizar la clasificación automática, se utilizaron dos técnicas de evaluación: Cross-Validation, que consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones (Payam Refaeilzadeh, Lei Tang, 2008) y Percentage Split que consiste en dividir un dataset en porcentajes de datos de prueba y de entrenamiento. Para esta fase de clasificación, se utilizó la aplicación java **Weka** que provee un set de herramientas Machine Learning con las cuales se realiza la clasificación del modelo de detección de ironía. Weka además trabaja con datasets en formato **.arff** el cual es generado en los pasos anteriormente descritos para cada tweet del dataset, en función de las características del modelo. De acuerdo a lo anterior, en este trabajo se utiliza la función **Percentage Split** de Weka que divide el dataset en porciones de evaluación y entrenamiento. En este trabajo se utiliza 70 % del dataset como datos de entrenamiento y 30 % como datos de prueba. Por otro lado, se realiza una nueva evaluación utilizando ahora la función **Cross-validation** utilizando 10 sets para la realización de pruebas cruzadas. En este trabajo, los dataset son conformados por elementos del corpus de

evaluación etiquetado en el Capítulo 4. Estas porciones del corpus contienen igual cantidad de tweets irónicos y no irónicos, de esta forma, se tiene un total de 1106 instancias, de las cuales, 555 corresponden a tweets etiquetados como irónicos y 555 a tweets no irónicos. De esta forma, por ejemplo, para realizar la función percentage split, se utilizaron 776 instancias para entrenamiento y 334 para pruebas, las cuales se componen de 50 % tweet irónicos y 50 % tweets no irónicos. Se generaron distintos dataset **.arff** para poder probar distintas combinaciones de atributos, la cuales entregan distintos rendimientos en función del corpus de evaluación. Este trabajo consideró las siguientes combinaciones:

(a) Agregar de forma **progresiva** los atributos partiendo de un solo atributo hasta probar con todos juntos. Esto se ilustra en la Figura 5.2
(b) Probar con cada atributo de forma **independiente**, para saber cual es el rendimiento individual de cada atributo.

Las combinaciones anteriores permiten probar, el rendimiento individual de cada atributo del modelo, con el fin de observar cuáles tienen mejores rendimientos y además analizar los rendimientos al combinar los atributos entre sí. En este trabajo, se consideró agregarlos de forma progresiva, partiendo desde uno hasta agregarlos todos juntos ya que de esta forma se pueden revisar las combinaciones entre atributos que entreguen los mejores rendimientos.

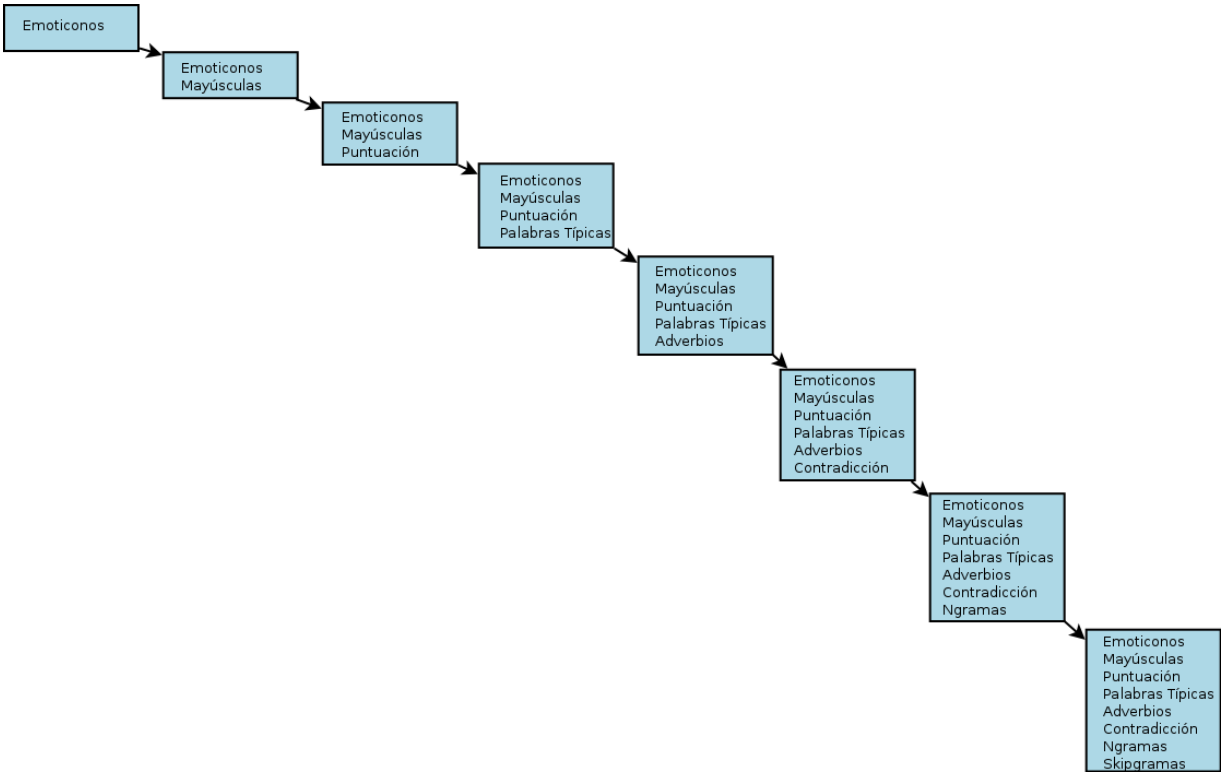


Figura 5.2: Esquema de agregación progresiva de los atributos.

Como se aprecia en la Figura 5.2, se comienza con el procesado para el primer grupo, que correspode al Uso de emoticonos. Esto genera el primer dataset en formato **.arff** compuesto del valor característica, que es calculado como se describió anteriormente y su clase, en este caso irónico o no irónico. Lo anterior, se realiza para cada elemento del corpus descrito anteriormente. Luego, para el segundo grupo, compuesto por las características Uso de emoticonos y Uso de signos de puntuación se repite el mismo proceso, hasta completar todos los grupos. En total, se procesan 9 datasets **.arff** distintos, que corresponden con los grupos a formar con los atributos del modelo, para cada elemento del corpus de evaluación.

Para la segunda combinación, se genera un dataset **.arff** para cada atributo de forma individual. A diferencia de la combinación anterior, se generan 9 datasets **.arff** pero con el cálculo de cada característica de forma individual, sin considerar las demás ni formar grupos entre ellas.

## 5.3. Conclusión del Capítulo

El modelo de detección de ironía propuesto en este trabajo está compuesto de 9 atributos o características, las cuales son: uso de emoticonos, uso signos de puntuación, uso de palabras en mayúscula, presencia de signos típicos de ironía, uso de adverbios no temporales, presencia de contradicción texto-emoticón, uso de adverbios temporales, análisis de ngramas y análisis de skipngramas. Estos atributos permiten analizar un texto y buscar patrones típicos de ironía, los cuales mediante fórmulas anteriomente descritas, permiten construir datasets en formato **.arff** y clasificar automáticamente un texto en irónico y no irónico. Este modelo utiliza el enfoque de machine learning llamado aprendizaje supervisado mediante la aplicación Weka, que recibe datasets en formato **.arff** y utiliza el algoritmo Naive Bayes para realizar clasificación automática de textos en clases. En el siguiente capítulo se exponen los resultados de la evaluación de este modelo descrito.

# Capítulo 6

# Resultados

En esta sección se presentan los resultados de la evaluación del modelo de detección de ironía propuesto en este trabajo. Se describen el experimento realizado y los resultados obtenidos a partir de la clasificación automática usando técnicas de Machine Learning con la herramienta Weka. Se utilizaron 3 métricas de clasificación (Olson y Delen, 2008):Precision, Recall y F-Measure.

Para simplificar los nombres de los atributos que componen el modelo de detección de ironía se utilizaron alias para cada uno de ellos, mediante los cuales son identificados en las Tablas y Gráficos del detalle de los resultados. Estos alias se describen en la Tabla 6.1.

| Atributo | Alias |
|---|---|
| Uso de emoticonos | Emoticonos |
| Uso signos de puntuación | Puntuación |
| Uso de palabras en mayúscula | Mayúsculas |
| Presencia de signos típicos de ironía | Signos Típicos |
| Uso de adverbios no temporales | Adv. No Temporal |
| Presencia de contradicción texto-emoticón | Adv. Temporal |
| Uso de adverbios temporales | Contradicción |
| Análisis de ngramas | Ngramas |
| Análisis de skipngramas | Skipngramas |

Tabla 6.1: Alias por nombre de atributo del modelo de detección de ironía.

## 6.1. Detalles de los Resultados

A continuación se presentan los resultados obtenidos al ir agregando progresivamente los atributos del modelo por cada una de las distintas métricas de evaluación utilizadas.

Para la etapa de agregación progresiva de atributos, se han agrupado los atributos de la siguiente forma:

- **Grupo 1:**
  - Emoticonos
- **Grupo 2**:
  - Emoticonos
  - Puntuación
- **Grupo 3**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
- **Grupo 4**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
  - Signos Típicos
- **Grupo 5**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
  - Signos Típicos
  - Adv. No Temporal
- **Grupo 6**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
  - Signos Típicos
  - Adv. No Temporal
  - Adv. Temporal
- **Grupo 7**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
  - Signos Típicos
  - Adv. No Temporal
  - Adv. Temporal
  - Contradicción
- **Grupo 8**:
  - Emoticonos
  - Puntuación
  - Mayúsculas
  - Signos Típicos

- Adv. No Temporal
- Adv. Temporal
- Contradicción
- Ngramas

▪ **Grupo 9**:

- Emoticonos
- Puntuación
- Mayúsculas
- Signos Típicos
- Adv. No Temporal
- Adv. Temporal
- Contradicción
- Ngramas
- Skipngramas

Por lo tanto, en las Tablas 6.2, 6.3, 6.6 , 6.7 los grupos numerados corresponden con la descripción anterior.

### 6.1.1. Primera Prueba: 70 % entrenamiento y 30 % evaluación

La Tabla 6.2 resume los valores de las métricas **Precision**, **Recall** y **F-Measure** para cada clase de las distintas etapas de agregación progresiva de atributos. Se pueden ver graficados estos resultados en las Figuras 6.1 y 6.2

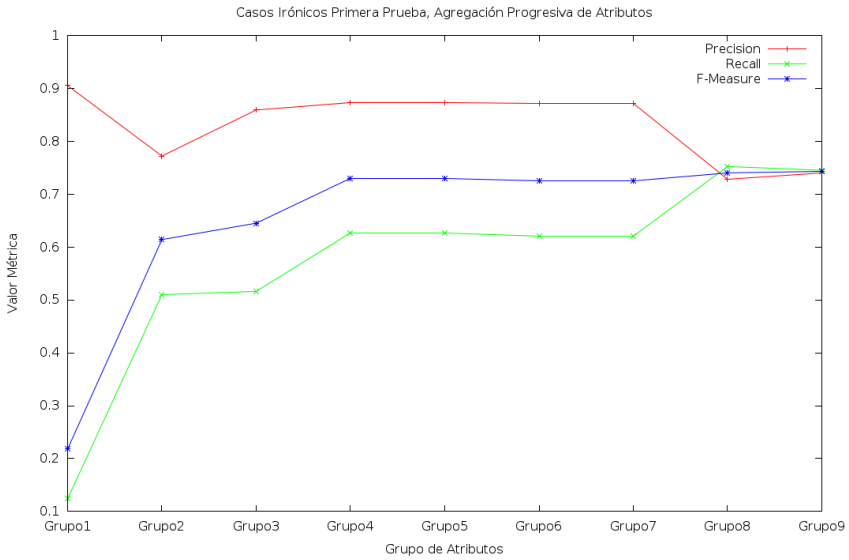


Figura 6.1: Rendimiento para casos irónicos por métricas primera prueba, con agregación progresiva de atributos.

| Etapa | Clase | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Grupo 1 | Irónico | 0,905 | 0,124 | 0,218 |
| | No Irónico | 0,569 | 0,989 | 0,722 |
| Grupo 2 | Irónico | 0,772 | 0,510 | 0,614 |
| | No Irónico | 0,675 | 0,872 | 0,761 |
| Grupo 3 | Irónico | 0,859 | 0,516 | 0,645 |
| | No Irónico | 0,692 | 0,927 | 0,792 |
| Grupo 4 | Irónico | 0,873 | 0,627 | 0,730 |
| | No Irónico | 0,743 | 0,922 | 0,823 |
| Grupo 5 | Irónico | 0,873 | 0,627 | 0,730 |
| | No Irónico | 0,743 | 0,922 | 0,823 |
| Grupo 6 | Irónico | 0,872 | 0,621 | 0,725 |
| | No Irónico | 0,740 | 0,922 | 0,821 |
| Grupo 7 | Irónico | 0,872 | 0,621 | 0,725 |
| | No Irónico | 0,740 | 0,922 | 0,821 |
| Grupo 8 | Irónico | 0,728 | 0,752 | 0,740 |
| | No Irónico | 0,782 | 0,760 | 0,771 |
| Grupo 9 | Irónico | 0,740 | 0,745 | 0,743 |
| | No Irónico | 0,781 | 0,777 | 0,779 |

Tabla 6.2: Resumen resultados clasificación primera prueba.

Además, la Tabla 6.3 resume los rendimientos generales de las etapas de agregación progresiva de atributos.

| Etapa | Rendimiento General |
|---|---|
| Grupo 1 | 59.0361 % |
| Grupo 2 | 70.4819 % |
| Grupo 3 | 73.7952 % |
| Grupo 4 | 78.6145 % |
| Grupo 5 | 78.6145 % |
| Grupo 6 | 78.3133 % |
| Grupo 7 | 78.3133 % |
| Grupo 8 | 75.6024 % |
| Grupo 9 | 76.2048 % |

Tabla 6.3: Resumen resultados generales clasificación primera prueba, agregación progresiva de atributos.

Se observa en la Tabla 6.2 que para el primer atributo (Grupo 1) se obtuvieron valores de

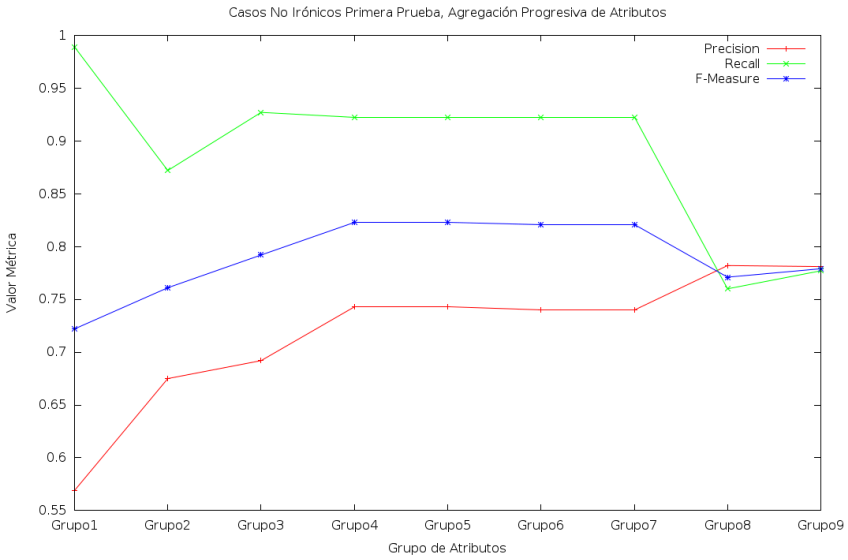


Figura 6.2: Rendimiento para casos no irónicos por métricas primera prueba, con agregación progresiva de atributos.

90,5 % de Precisión para las clases etiquetadas como irónicas y 56,9 % para las clases etiquetadas como no irónicas. Al agregar el segundo atributo (Grupo 2), los valores cambian a 77,2 % y 67,5 % de Precisión en las clases irónicas y no irónicas respecticamente. Con tres atributos (Grupo 3) agregados se obtienen 85,9 % y 69,2 % de Precisión en las clases irónicas y no irónicas. Mejora considerablemente el rendimiento en ambas clases al agregar 4 y 5 atributos (Grupos 4 y 5), obteniéndose para la clase irónica un rendimiento de 87,3 % de Precisión y 74,3 % de Precisión en las clases no irónicas. Resultados similares se obtienen al agregar 6 y 7 atributos con valores de Precisión de 87,2 % y 74 % en las clases irónicas y no irónicas. Finalmente, al agregar todos los atributos, se obtiene una baja en el rendimiento al obtenerse valores de Precisión de 74 % y 78,1 % en las clases irónicas y no irónicas.

Como se pueden observar en los resultados generales de esta primera prueba, descritos en la Tabla 6.3, la combinación que obtiene el mejor rendimiento del modelo es incorporando los atributos correspondientes a las características **Emoticonos, Mayúsculas, Signos de Puntuación, Palabras Típicas de Ironía y Adverbios** obteniéndose un rendimiento general de hasta un 78,6 %. El rendimiento tiende a disminuir con el uso de ngramas y aumentar levemente al agregar los skipngramas.

Para finalizar la primera prueba, se realizó el mismo procedimiento, pero con cada atributo de forma **independiente**, es decir, cada característica o atributo del modelo por separado. La Tabla 6.4 resume los resultados obtenidos para cada atributo independiente. Se muestran además en la Tabla 6.5 un resumen de los rendimientos generales de cada clasificación por atributo independiente. Finalmente, las Figuras 6.3 y 6.4 muestran los gráficos de estos resultados.

Se observa en la Tabla 6.4 que para el atributo **Emoticonos** se obtuvieron valores de 90,5 %

| Etapa | Clase | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Emoticonos | Irónico | 0,905 | 0,124 | 0,218 |
| | No Irónico | 0,569 | 0,989 | 0,722 |
| Puntuación | Irónico | 0,747 | 0,405 | 0,525 |
| | No Irónico | 0,635 | 0,883 | 0,738 |
| Mayúsculas | Irónico | 0,844 | 0,176 | 0,292 |
| | No Irónico | 0,580 | 0,972 | 0,727 |
| Signos Típicos | Irónico | 0,960 | 0,157 | 0,270 |
| | No Irónico | 0,580 | 0,994 | 0,733 |
| Adv. No Temporal | Irónico | 0,482 | 0,889 | 0,625 |
| | No Irónico | 0,660 | 0,184 | 0,288 |
| Adv. Temporal | Irónico | 0,872 | 0,621 | 0,725 |
| | No Irónico | 0,740 | 0,922 | 0,821 |
| Contradicción | Irónico | 0,459 | 0,961 | 0,622 |
| | No Irónico | 0,500 | 0,034 | 0,063 |
| Ngramas | Irónico | 0,724 | 0,719 | 0,721 |
| | No Irónico | 0,761 | 0,765 | 0,763 |
| Skipngramas | Irónico | 0,582 | 0,693 | 0,633 |
| | No Irónico | 0,687 | 0,575 | 0,626 |

Tabla 6.4: Resumen resultados clasificación primera prueba atributos independientes.

| Etapa | Rendimiento General |
|---|---|
| Emoticonos | 59.0361 % |
| Puntuación | 66.2651 % |
| Mayúsculas | 60.5422 % |
| Signos Típicos | 60.8434 % |
| Adv. No Temporal | 50.9036 % |
| Adv. Temporal | 46.0843 % |
| Contradicción | 46.0843 % |
| Ngramas | 74.3976 % |
| Skipngramas | 62.9518 % |

Tabla 6.5: Resumen resultados generales clasificación primera prueba, atributos independientes.

de Precisión para las clases etiquetadas como irónicas y 56,9 % para las clases etiquetadas como no irónicas. Para el atributo **Puntuación**, los valores cambian a 74,7 % y 63,5 % de Precisión en las clases irónicas y no irónicas respecticamente. El atributo **Mayúsculas** obtiene 84,4 % y 58 % de Precisión en las clases irónicas y no irónicas. Se obtienen rendimientos de Precisión de 96 % y 58 % para las clases irónicas y no irónicas con el atributo **Signos Típicos**, siendo

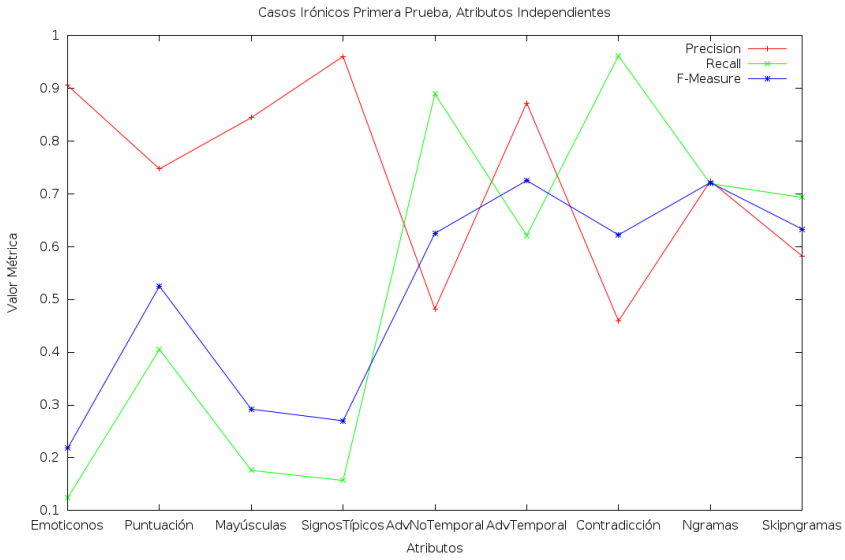


Figura 6.3: Rendimiento para casos irónicos por métricas primera prueba, con atributos independientes.

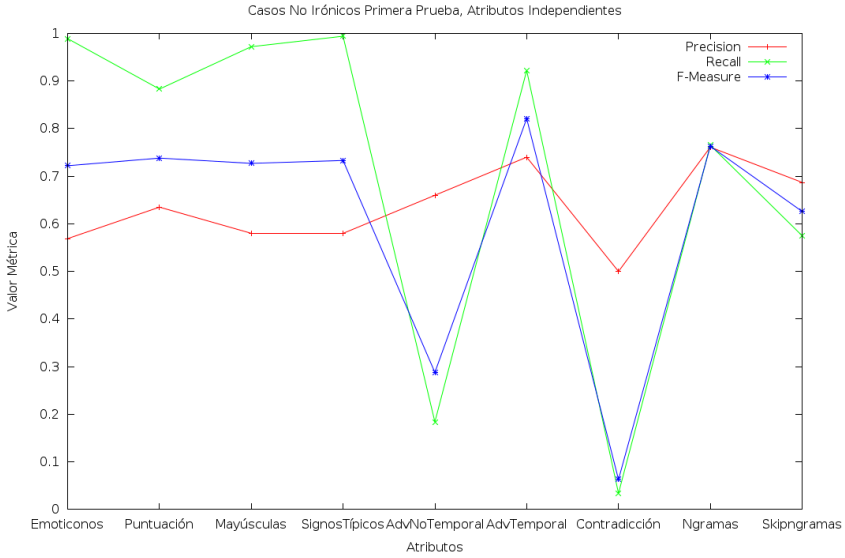


Figura 6.4: Rendimiento para casos no irónicos por métricas primera prueba, con atributos independientes.

el rendimiento para la clase irónica el más alto de todos los atributos individuales. Para los atributos **Adverbios No Temporales**, **Adverbios Temporales** se obtienen precisiones de 48,2 % y 87,2 % para las clases irónicas, además de 66 % y 74 % en las clases no irónicas. El

rendimiento más bajo se obtiene al evaluar el atributo **Contradicción** con valores de Precisión de 45,9 % y 50 % en la clases irónicas y no irónicas. Finalmente, en los atributos **Ngramas** y **Skipngramas** se obtienen valores de Precisión de 72,4 % y 58,2 % en la clase irónica y 76,1 % y 68,7 % en la clase no irónica. En terminos generales, observados en la Tabla 6.5, el único atributo con un rendimiento aceptable (por si solo) es el uno de **Ngramas**, el resto, con valores que oscilan entre los 46 y 66 % de Precisión, que son considerados rendimientos bajos.

### 6.1.2. Segunda Prueba: Fold Cross Validation (10 folds)

La segunda prueba se realizó bajo el enfoque de Fold Cross Validation. La Tabla 6.6 resume los valores de las métricas **Precision**, **Recall** y **F-Measure** para cada clase de las distintas etapas de agregación progresiva de atributos. Estos resultados están graficados en las Figuras 6.5 y 6.6

| Etapa | Clase | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Grupo 1 | Irónico | 0,923 | 0,108 | 0,194 |
| | No Irónico | 0,527 | 0,991 | 0,688 |
| Grupo 2 | Irónico | 0,795 | 0,448 | 0,573 |
| | No Irónico | 0,616 | 0,884 | 0,726 |
| Grupo 3 | Irónico | 0,862 | 0,430 | 0,574 |
| | No Irónico | 0,621 | 0,931 | 0,745 |
| Grupo 4 | Irónico | 0,876 | 0,497 | 0,634 |
| | No Irónico | 0,649 | 0,930 | 0,765 |
| Grupo 5 | Irónico | 0,877 | 0,505 | 0,641 |
| | No Irónico | 0,653 | 0,930 | 0,767 |
| Grupo 6 | Irónico | 0,877 | 0,503 | 0,639 |
| | No Irónico | 0,652 | 0,930 | 0,766 |
| Grupo 7 | Irónico | 0,877 | 0,503 | 0,639 |
| | No Irónico | 0,652 | 0,930 | 0,766 |
| Grupo 8 | Irónico | 0,831 | 0,709 | 0,765 |
| | No Irónico | 0,746 | 0,856 | 0,797 |
| Grupo 9 | Irónico | 0,831 | 0,702 | 0,761 |
| | No Irónico | 0,742 | 0,857 | 0,796 |

Tabla 6.6: Resumen resultados clasificación segunda prueba.

Además, la Tabla 6.7 resume los rendimientos generales de las etapas de agregación progresiva de atributos.

Se observan rendimientos bastante similares con la prueba 1. Se observa en la Tabla 6.6 que para el primer atributo (Grupo 1) se obtuvieron valores de 92,3 % de Precisión para las clases etiquetadas como irónicas y 52,7 % para las clases etiquetadas como no irónicas. Al agregar el segundo atributo (Grupo 2), los valores cambian a 79,5 % y 61,6 % de Precisión en las clases

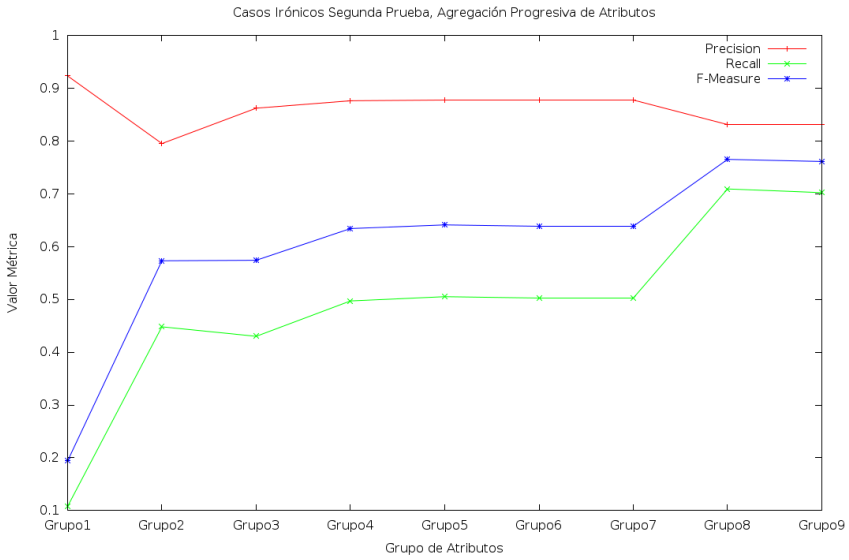


Figura 6.5: Rendimiento para casos irónicos por métricas segunda prueba, con agregación progresiva de atributos.

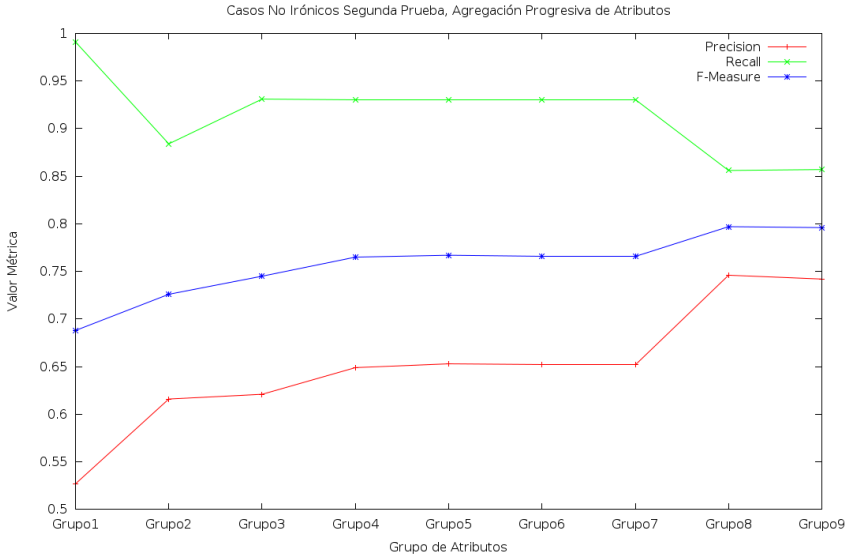


Figura 6.6: Rendimiento para casos no irónicos por métricas segunda prueba, con agregación progresiva de atributos.

irónicas y no irónicas respecticamente. Con tres atributos agregados se obtienen 86,2 % y 62,1 % de Precisión en las clases irónicas y no irónicas. Con cuatro atributos agregados (Grupo 4) se obtienen 87,6 % y 64,9 % de Precisión en las clases irónicas y no irónicas. Mejora considerablemente

| Etapa | Rendimiento General |
|---|---|
| Grupo 1 | 55.0136 % |
| Grupo 2 | 66.6667 % |
| Grupo 3 | 68.112 % |
| Grupo 4 | 71.364 % |
| Grupo 5 | 71.7254 % |
| Grupo 6 | 71.635 % |
| Grupo 7 | 71.635 % |
| Grupo 8 | 78.2294 % |
| Grupo 9 | 77.9584 % |

Tabla 6.7: Resumen resultados generales clasificación segunda prueba, agregación progresiva de atributos.

el rendimiento en ambas clases al agregar 5 y 7 atributos (Grupo 5 y 7), obtieniendose para la clase irónica un rendimiento de 87,7 % de Precisión y 65,3 % de Precisión en las clases no irónicas. Finalmente, al agregar todos los atributos (Grupo 9), se obtiene un rendimiento con valores de Precisión de 83,1 % y 74,2 % en las clases irónicas y no irónicas. A diferencia con la primera prueba la combinación que obtiene el mejor rendimiento del modelo es colocando los atributos correspondientes a las características **Emoticonos, Mayúsculas, Signos de Puntuación, Palabras Típicas de Ironía, Adverbios, Contradicción y Ngramas** obteniéndose un rendimiento general de hasta un 78,2 %, como se aprecia en la Tabla 6.7. El rendimiento tiende a aumentar con el uso de Ngramas y a disminuir levemente al agregar los Skipngramas.

Para finalizar la segunda prueba, se realizó el mismo procedimiento que en la primera prueba con cada atributo de forma **independiente**. Esto resultados se muestran en la Tabla 6.8 y se grafican en las Figuras 6.7 y 6.8.

Finalmente, la Tabla 6.9 resume los rendimientos generales de cada clasificación por atributo.

Se observa en la Tabla 6.9 que para el atributo **Emoticonos** se obtuvieron valores de 92,3 % de Precisión para las clases etiquetadas como irónicas y 52,7 % para las clases etiquetadas como no irónicas. Para el atributo **Puntuación**, los valores cambian a 73,9 % y 61,6 % de Precisión en las clases irónicas y no irónicas respecticamente. El atributo **Mayúsculas** obtiene 82,6 % y 53,6 % de Precisión en las clases irónicas y no irónicas. Se obtienen rendimientos de Precisión de 91,7 % y 52,9 % para las clases irónicas y no irónicas con el atributo **Signos Típicos**. Para los atributos **Adverbios No Temporales**, **Adverbios Temporales** se obtienen precisiones de 51 % y 48,4 % para las clases irónicas, además de 56,3 % y 48,5 % en las clases no irónicas. Al evaluar el atributo **Contradicción** se obtienen valores de Precisión de 92,3 % y 50,5 % en la clases irónicas y no irónicas. Finalmente, en los atributos **Ngramas** y **Skipngramas** se obtienen valores de Precisión de 82,9 % y 59,3 % en la clase irónica y 74,1 % y 59,6 % en la clase no irónica.

Se observan leves diferencias en comparación con la primera prueba, ya que los resultados generales observados en la Tabla 6.9 tuvieron una leve mejora. En terminos generales, el único atributo con un rendimiento aceptable (por si solo) sigue siendo el uno de **Ngramas**, el resto,

| Etapa | Clase | Precision | Recall | F-Measure |
|---|---|---|---|---|
| Emoticonos | Irónico | 0,923 | 0,108 | 0,194 |
| | No Irónico | 0,527 | 0,991 | 0,688 |
| Puntuación | Irónico | 0,739 | 0,481 | 0,583 |
| | No Irónico | 0,616 | 0,830 | 0,707 |
| Mayúsculas | Irónico | 0,826 | 0,163 | 0,272 |
| | No Irónico | 0,536 | 0,966 | 0,689 |
| Signos Típicos | Irónico | 0,917 | 0,119 | 0,211 |
| | No Irónico | 0,529 | 0,989 | 0,690 |
| Adv. No Temporal | Irónico | 0,510 | 0,875 | 0,644 |
| | No Irónico | 0,563 | 0,161 | 0,250 |
| Adv. Temporal | Irónico | 0,484 | 0,481 | 0,482 |
| | No Irónico | 0,485 | 0,487 | 0,486 |
| Contradicción | Irónico | 0,923 | 0,022 | 0,042 |
| | No Irónico | 0,505 | 0,998 | 0,671 |
| Ngramas | Irónico | 0,829 | 0,700 | 0,759 |
| | No Irónico | 0,741 | 0,856 | 0,794 |
| Skipngramas | Irónico | 0,593 | 0,600 | 0,597 |
| | No Irónico | 0,596 | 0,588 | 0,592 |

Tabla 6.8: Resumen resultados clasificación segunda prueba atributos independientes.

| Etapa | Rendimiento General |
|---|---|
| Emoticonos | 55.0136 % |
| Puntuación | 65.5827 % |
| Mayúsculas | 56.4589 % |
| Signos Típicos | 60.8434 % |
| Adv. No Temporal | 51.7615 % |
| Adv. Temporal | 48.4192 % |
| Contradicción | 51.0388 % |
| Ngramas | 77.7778 % |
| Skipngramas | 59.4399 % |

Tabla 6.9: Resumen resultados generales clasificación segunda prueba, atributos independientes.

con valores que oscilan entre los 48 y 65 % de Precisión, que son considerados rendimientos bajos.

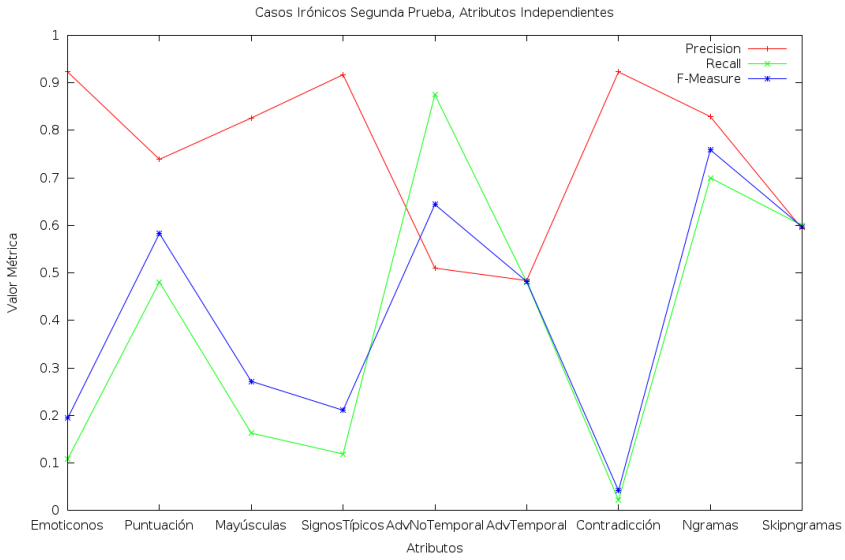


Figura 6.7: Rendimiento para casos irónicos por métricas segunda prueba, con atributos independientes.

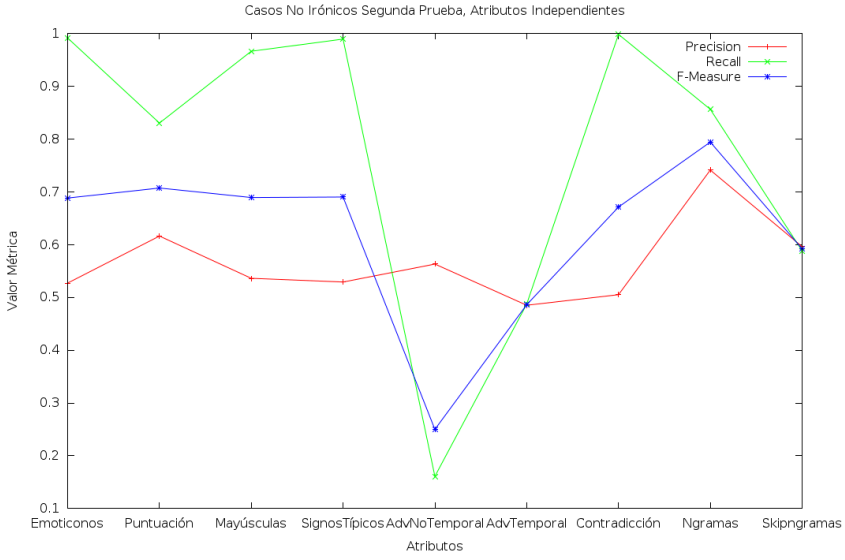


Figura 6.8: Rendimiento para casos no irónicos por métricas segunda prueba, con atributos independientes.

## 6.2. Conclusión del Capítulo

Se realizaron 2 pruebas para evaluar el modelo de detección de ironía, las cuales estuvieron basadas en enfoque Percentage Split y Fold Cross Validation. Estas pruebas permiten analizar resultados en función de la rigurosidad propia de cada enfoque. A continuación se presentan comparaciones entre pruebas.

### 6.2.1. Comparación de Pruebas con Atributos Agregados Progresivamente

Al comparar los gráficos de las Figuras 6.1 y 6.2 que corresponden a los resultados de la primera prueba para casos irónicos y no irónicos respectivamente, con los de las Figuras 6.5 y 6.6 que corresponden a los resultados de la segunda prueba, con caso irónicos y no irónicos respectivamente se puede observar que existen diferencias bastante pequeñas entre ambos enfoques. Como se puede observar en la Tabla 6.10 para casos irónicos, considerando el primer atributo (Grupo 1) se obtuvo en la primera prueba una Precisión de 90,5 % y en la segunda prueba 92,3 %, generandose una diferencia de 1,8 % a favor del enfoque Fold Cross Validation. Para los demás grupos de atributos, se obtuvieron en la primera prueba para el Grupo 2 un 77,2 % de Precisión y en la segunda prueba 79,5 % con una diferencia de 2,3 % a favor del enfoque Fold Cross Validation. El Grupo 3 obtuvo una Precisión en la primera prueba de 85,9 % y en la segunda prueba 86,2 % con una diferencia de 0,3 % a favor del enfoque Fold Cross Validation. Los Grupos 4 al 7 obtuvieron diferencias de entre 0,3 y 0,5 % a favor del enfoque Fold Cross Validation. Los grupos de atributos que obtuvieron mayores diferencias fueron el Grupo 8 y el Grupo 9, con Precisiones de 72,8 y 74 % en la primera prueba y 83,1 ambas en la segunda, logrando diferencias de 10,3 y 9,1 % a favor del enfoque Fold Cross Validation. Por el contrario, como se observa en la Tabla 6.11 para los casos no irónicos, se obtuvieron resultados con mayores diferencias. Por ejemplo, para los grupos de atributos 4 y 7 se obtuvieron diferencias de 9,4 % y 8,8 % a favor del enfoque Percentage Split. El resto, con diferencias entre 3,6 y 7,1 % a favor del enfoque Percentage Split.

| Grupos | Precisión Prueba 1 | Precisión Prueba 2 | Diferencia |
|---|---|---|---|
| Grupo 1 | 0,905 | 0,923 | -0,018 |
| Grupo 2 | 0,772 | 0,795 | -0,023 |
| Grupo 3 | 0,859 | 0,862 | -0,003 |
| Grupo 4 | 0,873 | 0,876 | -0,003 |
| Grupo 5 | 0,873 | 0,877 | -0,004 |
| Grupo 6 | 0,872 | 0,877 | -0,005 |
| Grupo 7 | 0,872 | 0,877 | -0,005 |
| Grupo 8 | 0,728 | 0,831 | -0,103 |
| Grupo 9 | 0,740 | 0,831 | -0,091 |

Tabla 6.10: Resumen comparación de Precisiones entre prueba 1 y prueba 2 para casos irónicos, con agregación progresiva de atributos.

| Grupos | Precisión Prueba 1 | Precisión Prueba 2 | Diferencia |
|---|---|---|---|
| Grupo 1 | 0,569 | 0,527 | 0,042 |
| Grupo 2 | 0,675 | 0,616 | 0,059 |
| Grupo 3 | 0,692 | 0,621 | 0,071 |
| Grupo 4 | 0,743 | 0,649 | 0,094 |
| Grupo 5 | 0,743 | 0,653 | 0,090 |
| Grupo 6 | 0,740 | 0,652 | 0,088 |
| Grupo 7 | 0,740 | 0,652 | 0,088 |
| Grupo 8 | 0,782 | 0,746 | 0,036 |
| Grupo 9 | 0,781 | 0,742 | 0,039 |

Tabla 6.11: Resumen comparación de Precisiones entre prueba 1 y prueba 2 para casos no irónicos, con agregación progresiva de atributos.

### 6.2.2. Comparación de Pruebas con Atributos Independientes

Comparando los gráficos de las Figuras 6.3 y 6.4 que corresponden a los resultados de la primera prueba para casos irónicos y no irónicos respectivamente, con los de las Figuras 6.7 y 6.8 que corresponden a los resultados de la segunda prueba, con caso irónicos y no irónicos respectivamente se puede observar que al igual que en los casos anteriormente descritos (con grupos progresivos de atributos) aparecen leves diferencias. Observando la tabla 6.12 que corresponde a las Precisiones de los casos irónicos, con atributos independientes, se aprecia que el valor máximo de diferencia de Precisión a favor del enfoque Fold Cross Validation que corresponde al atributo Contradicción con un 46,4 % de diferencia. El resto de atributos se mantiene entre 1,8 y 10,5 % de diferencia, que corresponden a los atributos Mayúsculas y Ngramas respectivamente. Las diferencias de Precisión a favor de Percentage Split están comprendidas entre 0,8 y 38,6 %, correspondiendo a Puntuación y Adv. Temporal. Por otra parte, para los casos no irónicos, se observa en la tabla 6.13 que los valores de diferencia entre Precisiones de ambas pruebas casi en su totalidad son a favor de Percentage Split, con valores entre 2 y 25 % (Ngramas y Adv. Temporal). La única diferencia de Precisión a favor de Fold Cross Validation es de un 0,5 % y corresponde a Contradicción.

En terminos generales, considerando que el enfoque Fold Cross Validation es más riguroso que Percentage Split, se observan diferencias menores entre ambas pruebas, lo que permite contar con resultados generales estadísticamente confiables (Payam Refaeilzadeh, Lei Tang, 2008). A continuación, en el próximo capítulo se revisarán temas de discusión relacionados al trabajo realizado y a la luz de los resultados expuestos en este capítulo.

| Atributos | Precisión Prueba 1 | Precisión Prueba 2 | Diferencia |
|---|---|---|---|
| Emoticonos | 0,905 | 0,923 | -0,018 |
| Puntuación | 0,747 | 0,739 | 0,008 |
| Mayúsculas | 0,844 | 0,826 | 0,018 |
| Signos Típicos | 0,960 | 0,917 | 0,043 |
| Adv. No Temporal | 0,482 | 0,510 | -0,028 |
| Adv. Temporal | 0,872 | 0,484 | 0,386 |
| Contradicción | 0,459 | 0,923 | -0,464 |
| Ngramas | 0,724 | 0,829 | -0,105 |
| Skipngramas | 0,582 | 0,593 | -0,011 |

Tabla 6.12: Resumen comparación de Precisiones entre prueba 1 y prueba 2 para casos irónicos, atributos independientes.

| Atributos | Precisión Prueba 1 | Precisión Prueba 2 | Diferencia |
|---|---|---|---|
| Emoticonos | 0,569 | 0,527 | 0,042 |
| Puntuación | 0,635 | 0,616 | 0,019 |
| Mayúsculas | 0,580 | 0,536 | 0,044 |
| Signos Típicos | 0,580 | 0,529 | 0,051 |
| Adv. No Temporal | 0,660 | 0,563 | 0,097 |
| Adv. Temporal | 0,740 | 0,485 | 0,255 |
| Contradicción | 0,500 | 0,505 | -0,005 |
| Ngramas | 0,761 | 0,741 | 0,020 |
| Skipngramas | 0,687 | 0,596 | 0,091 |

Tabla 6.13: Resumen comparación de Precisiones entre prueba 1 y prueba 2 para casos no irónicos, atributos independientes.

# Capítulo 7

# Discusión

A continuación se presenta la discusión de los resultados del experimento realizado.

En términos generales, nuestro modelo es capaz de detectar ironía en textos con un rendimiento de aproximadamente 78 % considerando la aplicación en conjunto de todos los atributos o características que componen el modelo. El resultado es aceptable en comparación a todos los trabajos revisados en esta investigación y más aún considerando la escasa o nula existencia de trabajos de este tipo para el idioma español.

Es interesante el análisis que surge a la luz de los resultados del experimento, ya que obtenemos rendimientos bastante distintos al ir agregando a los atributos en forma **progresiva** e **independiente**. Cuando los agregamos de forma **progresiva**, podemos observar claramente que nuestro modelo obtiene un mejor comportamiento al funcionar con 5 y 8 atributos, es decir con los atributos **{emoticonos, mayúsculas, puntuación, palabras típicas, adverbios, contradicción y ngramas}** obteniendo un rendimiento de hasta un 78 %. El peor rendimiento se obtiene al utilizar solo el primer atributo, con un 55 %.

Por otra parte, al realizar las mismas pruebas con los atributos de forma **independiente** se puede observar que el único atributo que por si solo obtiene el mejor rendimiento general es el **uso de ngramas** con un 77 %. Los demás atributos oscilan entre los 40 y 50 % de rendimiento general, valores considerados bajos. Probablemente, se puedan obtener resultados distintos probando distintas combinaciones o permutaciones de atributos del modelo entre si.

En definitiva, nuestro modelo de detección de ironía tiene un rendimiento general de un 78 %, y no todos los atributos o características aportan de la misma forma al rendimiento del modelo presentado y tienden a complementarse entre si, es decir, no funcionan de mejor manera de forma separada.

Por otro lado, en relación al marco teórico, la mayoría de los trabajos revisados, alrededor de 8 modelos distintos de detección de ironía y sarcasmo toman como base principal el trabajo de (Reyes et al., 2013), debido a que este es un modelo multidimensional que abarca aspectos linguísticos, semánticos y psicológicos. Lo anterior ha generado que la mayoría de los trabajos formulados hasta ahora tomen con base el mismo conjunto de características, y les asignen distintos nombres solamente o bien las adapten en otros contextos. La mayoría de aquellos trabajos han obtenido rendimientos considerados buenos entre los cuales se incluye nuestro trabajo, que

además introdujo 2 nuevas características obteniendo buenos resultados con la llamada **Signos típicos de ironía** y no tan buenos con **Contradicción Texto-Emoticón**.

Referente al corpus de evaluación creado en este trabajo, creemos que se necesitan mejoras a realizar. El trabajo podría mejorar si se consiguiera la participación de un número mayor de evaluadores en la encuesta y que estos pudiesen ser caracterizados, por ejemplo, en usuarios de redes sociales. De esa forma se podría diversificar más aún el conocimiento y la evaluación final de determinar la ironía en un texto. Por otro lado, existe cierta dificultad aún en las personas en determinar si un texto es o no irónico ya que aún entre humanos es difícil decidir o detectar ironía en un texto.

Creemos que se podría mejorar el corpus extendiendo el número de tweets con cifras sobre los 20.000, ya que el corpus más utilizado en los modelos revisados en este trabajo contiene 40.000 tweets (Reyes et al., 2013). Además sería un corpus más robuto si se añadieran más temas o dominios de diversas temáticas, no solo relacionadas con política y sociedad, debido a que lo más probable es que en otras áreas de dominio opinen personas distintas, con formas de escribir distintas y por ende, expresen la ironía de forma también diferente.

Como la mayoría de los tweets se recopilaron con temáticas relevantes para Chile, existieron muchos elementos descartados por ser mensajes ilegibles y con muchas faltas de ortografía. En ese sentido, el procesamiento del corpus podría mejorar si se aplican nuevas y mejores técnicas de pre-procesado enfocadas en mejorar la calidad de los tweets recogidos, en función de la ortografía y modísmos propios de un determinado país.

Nuestro modelo está basado en las características más utilizadas de todos los modelos revisados en este trabajo. Pudimos comprobar que muchas de esas características si son aplicables al idioma español con resultados aceptables. Las 2 características que introdujimos no tuvieron el éxito esperado. Por un lado, la característica **Signos típicos de ironía** influyó positivamente en el experimento al mejorar el rendimiento hacia un 87,3 %, sin embargo, la característica **Contradicción texto-emoticón** no logró perjudicar ni mejorar los resultados. Creemos que se debe que en este corpus no hay bastantes instancias que contengan texto y emoticones con contradicciones entre si. Probablemente, al cambiar de corpus, se puedan lograr resultados distintos para estas nuevas características. Utilizamos para el experimento las configuraciones más usadas por los trabajos hasta ahora, es decir, clasificación usando **Naive Bayes**. Quizás podríamos obtener resultados distintos cambiando el algoritmo de clasificación y sería interesante realizar una comparativa de cambios de rendimientos entre distintos algoritmos de clasificación o agregando de forma alternada los atributos que componen nuestro modelo.

# Capítulo 8

# Trabajos Futuros

Basandonos en todo lo realizado en este trabajo y lo expuesto en la sección de discusión, recomendamos los siguientes trabajos futuros.

- Crear un corpus de evaluación en idioma español considerando distintas temáticas de paises de habla hispana con la finalidad de juntar las distintas versiones del idioma español buscando características similares. ¿Qué pasaría con nuestro modelo de detección de ironía si el corpus de evaluación estuviera compuesto no solo de temáticas de opinión de Chile?
- Mejorar el corpus de este trabajo aumentando su número y las temáticas de dominio. Además, se podría aumentar el número de evaluadores para obtener mejor certeza de opiniones respecto a si un tweet es o no irónico.
- Se pueden crear nuevos métodos o funciones que ayuden a mejorar la calidad ortográfica de los textos, ya que muchas veces esto depende del país e idioma en que se escriban los textos. De esta forma, contando con métodos que puedan mejorar las etapas de pre-procesado del corpus se podrían analizar cualquier tipo de texto sin importar su calidad ortográfica.
- Observamos que muchos tweets descartados en la etapa de filtrado manual eran candidatos por ejemplo a ser tweets irónicos, pero se descartaban debido a que iban acompañado de una imágen. En la mayoría de los casos, estas imágenes eran recurrentes en un contexto determinado, por ejemplo, un "meme" [1] o una foto relacionada a un trending topic (tema popular de Twitter). Sería interesante incluír en otro modelo de detección de ironía el análisis de estas imágenes para determinar si un mensaje es o no irónico.
- Se podrían crear nuevos experimentos con el mismo modelo y probar rendimientos al combinar o permutar las características del modelo entre sí. De esta forma, se podrían encontrar rendimientos distintos en las distintas combinaciones o permutaciones de atributos entre sí.
- Sería interesante desarrollar una aplicación que permita parametrizar la evaluación, agregando y quitando atributos para analizar los distintos rendimientos que se podrían obtener con cada atributo del modelo o características nuevas que pudiesen agregarse.
- Complementar el modelo con nuevas características y revisar si se obtienen los mismos resultados, o si son mejores o peores. Sin duda que el modelo puede ser mejorado y esto

[1] Meme: Imagen humoristica que se utiliza en internet.: http://www.jornada.unam.mx/2014/07/08/cultura/a07n1cul

sería posible a través de la incorporación de nuevos atributos o características.

- Se podría cambiar el algoritmo de clasificación, por ejemplo, utilizar árboles de decisión o SVM en lugar de Naive Bayes. Posiblemente se obtengan resultados distintos y sería interesante poder determinar qué cosas cambian.
- Siguendo la idea anterior, se podría realizar una evaluación de algoritmos usando un set conformado por distintos clasificadores y analizar con cuales se obtienen mejores y peores resultados.
- Desarrollar una API para Twitter que permita recopilar grandes cantidades de tweets y descarte automáticamente retweets, conversaciones, anuncios publicitarios, etc. De esta forma, se podrían construir corpus de tweets con mayor facilidad.

# Capítulo 9

# Conclusiones

En este capítulo se presentan las conclusiones obtenidas a partir de la realización de este trabajo. Primero, se muestran el cumplimiento de la hipótesis y objetivos de este trabajo y luego, las conclusiones generales.

## 9.1. Verificación de Hipótesis y Objetivos

En el capítulo 2 se presentó el objetivo general del trabajo que era: *"Proponer un modelo que detecte ironía en textos, en idioma español, a partir de propuestas que utilizan Machine Learning y de análisis lexicón"*. A continuación, se muestran los objetivos específicos y la forma en que se dió cumplimiento a estos.

- *"Realizar una revisión sistemática de literatura respecto a enfoques actuales de detección de ironía en textos, sus datasets, resultados y aplicaciones":* Se realizó una revisión sistemática de literatura, mediante la cual se pudieron runir los principales modelos de detección de ironía y sarcasmo. Además, se analizaron diversos trabajos que trataban con los conceptos básicos de Minería de Opinión, Análisis de Sentientos, Ironía, Sarcasmo, Aprendizaje Supervisado y técnicas de Machine Learning, entre otros. Esta revisión sistemática permitió contar con un marco teórico de acuerdo al estado del arte en la materia de estudio.
- *"Crear un corpus de evaluación formado por mensajes de Twitter en español" :* Se creó un corpus de evaluación compuesto de 2.200 tweets etiquetados como no irónicos y 555 tweets etiquetados como irónicos. Estos tweets fueron etiquetados por evaluadores humanos, los cuales analizaron porciones del corpus. Este corpus está disponible en formato **.sql** y fue utilizado para evaluar el modelo de detección de ironía propuesto.
- *"Identificar las características más relevantes de los modelos de detección de ironía encontrados en la revisión sistemática de literatura y seleccionar aquellas que serán incluidas en el nuevo modelo a proponer":* Una vez completada la revisión sistemática se revisaron en detalle los trabajos encontrados que corresponden a modelos de detección de ironía. Además, se incluyeron trabajos que tratan la detección de sarcasmo, los cuales permitieron complementar los modelos de detección de ironía. Todos estos trabajos permitieron conformar un conjunto de características similares, las cuales, debido a su rendimiento y

frecuencia de uso en los distintos modelos existentes, se incluyeron como parte principal de este nuevo modelo de detección de ironía. Finalmente, se introdujeron dos nuevas características adicionales a las existentes: Contradicción Texto-Emoticón y Uso de Signos Típicos de Ironía.

- *"Implementar el nuevo modelo de detección de ironía en un prototipo de aplicación desarrollada en una plataforma a definir":* Se implementó una aplicación en la plataforma Python que realiza las tareas de procesado y pre-procesado de cada característica del modelo de detección de ironía para cada elemento del corpus de evaluación. Esta aplicación permite generar datasets en formato .arff, los cuales son datos de entrada de la aplicación Weka, que realiza clasificación automática usando un algoritmo Naive Bayes. De esta forma, mediante ambas aplicaciones, se puede contar con la evaluación del modelo de detección de ironía y obtener resultados para análisis. El detalle de esta implementación se encuentra en el Apéndice B.
- *"Validar la propuesta mediante un experimento de evaluación":* Se diseñó un experimento que consistió en dos pruebas, con distintos enfoques cada una. La primera prueba utilizó el enfoque de Percentage Split, que dividió cada dataset en datos de prueba y datos de entrenamiento. Esto se realizó para dos combinaciónes: agregación progresiva de atributos, en forma de grupos y agregación individual de atributos. La segunda prueba consistió en un enfoque de Fold Cross Validation, usando 10 folds, y del mismo modo que la primera, usó las mismas dos combinaciones de atributos. Este experimento permitió evaluar el nuevo modelo y obtener rendimientos generales de hasta un 78 %.
- *"Analizar de forma crítica y comparativa los resultados obtenidos y emitir conclusiones":* Se presentaron análisis críticos y se compararon los distintos resultados obtenidos de las dos pruebas realizadas. Estos resultados presentan un aporte para continuar líneas de trabajo futuras relacionadas a detección de ironía en textos y además para seguir perfeccionando el nuevo modelo presentado.

De esta forma, mediante estos objetivos específicos se cumplió el objetivo general descrito anteriormente, ya que se propuso un nuevo modelo que efectivamente funciona para el idioma español, debido a que el corpus de evaluación lo componen textos en español. Las propuestas con las cuales se formó la base del nuevo modelo efectivamente utilizan Machine Learning y Análisis Lexicón. Además, se pudo probar la hipótesis descrita en el capítulo 2 *"Es posible proponer un modelo para la detección de ironía en textos en español, basado en enfoques propuestos"* ya que efectivamente, se pudo proponer un nuevo modelo para la detección de ironía en textos en español, que fue basada en los enfoques actualmente propuestos.

## 9.2. Conclusiones Generales

Mediante este trabajo se propuso un nuevo modelo de detección automática de ironía en textos escritos en español, a partir de los enfoques estudiados en una revisión sistemática de literatura. En la cual se revisaron artículos relacionados con la detección de ironía, sarcasmo y minería de opinión. La revisión sistemática entregó trabajos importantes relacionados con detección de ironía y también de sarcasmo. Nuestro modelo de detección de ironía tomó como

base las características de ironía más utilizadas en los modelos revisados, además de introducir dos nuevas, quedando conformado finalmente por los atributos: **Uso de emoticonos, Uso de Mayúsculas, Uso de Signos de Puntuación, Uso de Palabras típicas de ironía, Uso de Adverbios temporales y no temporales, Análisis de Contradicción Texto-Emoticón, Análisis de ngramas y Análisis de skipgramas**. Para probar nuestro modelo creamos un corpus de evaluación de aproximadamente 3.021 tweets los cuales fueron analizados por evaluadores humanos quedando un total de 2.700 tweets etiquetados como irónicos y no irónicos con los cuales pudimos evaluar el modelo presentado. Para el experimento de evaluación utilizamos el algoritmo Naive Bayes realizando dos pruebas distintas, una usando configuración de 70 % datos de entrenamiento y 30 % de prueba y otra usando enfoque Fold Cross Validation. En ellas se obtuvieron resultados de hasta 87,3 % de precisión para casos irónicos y 78 % de precisión para casos no irónicos y un 78 % de rendimiento general. A la luz de estos resultados, concluimos que nuestro modelo entrega resultados prometedores en comparación a la mayoría de los trabajos revisados, más aún considerando que es uno de los primeros modelos de detección de ironía en español. Lo anterior abre bastantes posibilidades de investigación futura, a nivel de creación de un corpus más perfecto y de incluír nuevas características en el modelo de detección de ironía.

# Referencias


Rajeev Arora y Srinath Srinivasa. A faceted characterization of the opinion mining landscape. *2014 6th International Conference on Communication Systems and Networks, COMSNETS 2014*, págs. 1–6, 2014. doi:10.1109/COMSNETS.2014.6734936.

Cristina Bosco, Viviana Patti, Andrea Bolioli, y Departamento de Informatica. Developing Corpora for Sentiment Analysis : The Case of Irony and Senti – TUT ( Extended Abstract ). *Ijcai2015*, (Ijcai):4158–4162, 2015.

P Carvalho, L Sarmento, M Silva, y E de Oliveira. Clues for detecting irony in user-generated contents: oh...!! It's "so easy"; -). *In: TSA '09: proceeding of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion.*, págs. 53–56, 2009. doi: 10.1145/1651461.1651471.

Basilis Charalampakis, Dimitris Spathis, Elias Kouslis, y Katia Kermanidis. Detecting Irony on Greek Political Tweets : A Text Mining Approach. *Proceeding EANN '15 Proceedings of the 16th International Conference on Engineering Applications of Neural Networks (INNS)*, 2015.

Herbert H Clark y Richard J Gerrig. On the Pretense Theory of Irony. *Journal of Experimental Psychology: General*, 113(1):121–126, 1984.

Thomas Dietterich, Christopher Bishop, David Heckerman, Michael Jordan, y Michael Kearns. Introduction to Machine Learning Second Edition Adaptive Computation and Machine Learning. 2012.

Erik Forslid y Niklas Wikén. Automatic irony- and sarcasm detection in Social media. *Student thesis Master Programme in Engineering Physics*, 2015.

R. W. Gibbs. Irony in talk among friends. *Metaphor and symbol*, págs. 5–27, 2000.

Irazu Hernandez-Farıas, Jose Benedi, y Paolo Rosso. Applying Basic Features from Sentiment Analysis for Automatic Irony Detection. 9117:121–129, 2015. doi:10.1007/978-3-319-19390-8. URL http://link.springer.com/10.1007/978-3-319-19390-8.

B. Levin J. L. Fleiss y M. C. Paik. Statistical methods for rates and proportions. 2013.

Micheline Kamber Jiawei Han y Jian Pei. *Data Mining—Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2006*. 2011. URL http://hanj.cs.illinois.edu/bk3/.

Vlado Kešelj, Fuchun Peng, Nick Cercone, y Calvin Thomas. Pacific Association for Computational Linguistics N-GRAM-BASED AUTHOR PROFILES FOR AUTHORSHIP ATTRIBUTION.

Barbara Kitchenham. Procedures for Performing Systematic Reviews. *Keele University Technical Report TR/SE-0401*, 2004. URL http://www.inf.ufsc.br/{~}aldo.vw/kitchenham.pdf.

Lingpeng Kong y Likun Qiu. Formalization and rules for recognition of satirical irony. *Proceedings - 2011 International Conference on Asian Language Processing, IALP 2011*, págs. 135–138, 2011. doi:10.1109/IALP.2011.14.

Walter A. Koza, Pedro Alfaro-Faccio, y Ricardo Martínez Gamboa. Deteccion automatica de chilenismos verbales a partir de reglas morfosintacticas. Resultados preliminares. *Procesamiento de Lenguaje Natural*, (54):69–76, 2015.

Edwin Lunando y Ayu Purwarianti. Indonesian social media sentiment analysis with sarcasm detection. *2013 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, págs. 195–198, 2013. doi:10.1109/ICACSIS.2013.6761575.

Rada Mihalcea y Carlo Strapparava. LEARNING TO LAUGH (AUTOMATICALLY): COMPUTATIONAL MODELS FOR HUMOR RECOGNITION. *Computational Intelligence*, 22(2):126–142, 2006. ISSN 0824-7935. doi:10.1111/j.1467-8640.2006.00278.x. URL http://doi.wiley.com/10.1111/j.1467-8640.2006.00278.x.

Aminu Muhammad, Nirmalie Wiratunga, Robert Lothian, Gizem Gezici, Berrin Yanikoglu, Dilek Tapucu, y Yücel Saygın. Advances in Social Media Analysis. *Studies in Computational Intelligence*, 602:87–104, 2015. ISSN 1860949X. doi: 10.1007/978-3-319-18458-6. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84930965832{&}partnerID=tZOtx3y1{%}5Cnhttp://www.scopus.com/inward/record.url?eid=2-s2.0-84930959792{&}partnerID=tZOtx3y1.

Prateek Nagwanshi y C. E. Veni Madhavan. Sarcasm detection using sentiment and semantic features. *KDIR 2014 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, págs. 418–424, 2014. doi:10.5220/0005153504180424. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84909969590{&}partnerID=tZOtx3y1.

David L. Olson y Dursun Delen. *Advanced Data Mining Techniques*, tomo 1. 2008. URL http://lib.mdp.ac.id/ebook/KaryaUmum/Advanced-Data-Mining-Techniques.pdf.

Huan Liu Payam Refaeilzadeh, Lei Tang. Cross-Validation. págs. 532–538, 2008.

Veronica Perez-Rosas, Carmen Banea, y Rada Mihalcea. Learning Sentiment Lexicons in Spanish. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, págs. 3077–3081, 2012. doi:10.1.1.383.5959. URL http://aclasb.dfki.de/nlp/bib/L12-1645.

Antonio Reyes y Paolo Rosso. Mining Subjective Knowledge from Customer Reviews: A Specific Case of Irony Detection. *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, (1975):118–124, 2011. URL http://dl.acm.org/citation.cfm?id=2107653.2107668.

Antonio Reyes y Paolo Rosso. Making objective decisions from subjective data: Detecting irony in customer reviews. *Decision Support Systems*, 53(4):754–760, 2012. ISSN 01679236. doi: 10.1016/j.dss.2012.05.027. URL http://dx.doi.org/10.1016/j.dss.2012.05.027.

Antonio Reyes y Paolo Rosso. On the difficulty of automatically detecting irony: beyond a simple case of negation. *Knowledge and Information Systems*, págs. 1–20, 2013. ISSN 02191377. doi: 10.1007/s10115-013-0652-8.

Antonio Reyes, Paolo Rosso, y Davide Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data and Knowledge Engineering*, 74:1–12, 2012. ISSN 0169023X. doi:10.1016/j.datak.2012.02.005. URL http://dx.doi.org/10.1016/j.datak.2012.02.005.

Antonio Reyes, Paolo Rosso, y Tony Veale. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, 47(1):239–268, 2013. ISSN 1574020X. doi: 10.1007/s10579-012-9196-x.

B. Seerat y F. Azam. Opinion mining: Issues and challenges (a survey). 2012.

Sperber y Wilson. Postface to the second edition of relevance: Communication and cognition. 1995.

Chih-Fong Tsai y Chih-Fong. Bag-of-Words Representation in Image Annotation: A Review. *ISRN Artificial Intelligence*, 2012:1–19, 2012. ISSN 2090-7443. doi:10.5402/2012/376804. URL http://www.hindawi.com/journals/isrn/2012/376804/.

Byron C. Wallace. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483, 2015. ISSN 02692821. doi:10.1007/s10462-012-9392-5.

Zelin Wang, Zhijian Wu, Ruimin Wang, y Yafeng Ren. Web Information Systems Engineering – WISE 2015. 9419:332–336, 2015. ISSN 16113349. doi:10.1007/978-3-319-26187-4. URL http://link.springer.com/10.1007/978-3-319-26187-4.

Cynthia Whissell. Using the Revised Dictionary of Affect in Language to quantify the emotional undertones of samples of natural language. *Psychological reports*, 105(2):509–21, 2009. ISSN 0033-2941. doi:10.2466/PR0.105.2.509-521. URL http://www.ncbi.nlm.nih.gov/pubmed/19928612.

D. Wilson y D. Sperber. On verbal irony. 87(1):53–76, 1992.

# Apéndice A

# Anexos del Capítulo 5

## A.1. Lista Adverbios

- Lista de adverbios no temporales: aproximadamente, alrededor, casi, cerca, entonces, incluso, más, menos, no, sólo, simplemente, sencillamente, todavía, aún, virtualmente, mas, solo, todavia, aun, aunque, ahora, pero, con todo, de allí, no obstante, sin embargo, de alli, a pesar de, más o menos, por lo tanto, punto menos que, mas o menos
- Lista de adverbios para atributo temporales: abruptamente, improvisadamente, después, luego, ahora, inesperadamente, recientemente, recién, últimamente, pronto, ayer, desde, repentino, súbito, rápido, precipitado, repentinamente, mañana, despues, recien, ultimamente, subito, rapido, más tarde, sin avisar, en breve, desde entonces, de repente, cuando sea, cuando quiera, mas tarde, de la nada, dentro de poco

# Apéndice B

# Anexos de la Implementación

Este anexo describe el entorno de desarrollo y los métodos utilizados para implementar el modelo de detección de ironía mediante una aplicación software. Esta aplicación permite realizar tareas de pre-procesamiento, procesamiento, construcción de dataset formato .arff y clasificación.

## B.1. Entorno de Desarrollo

### B.1.1. Características del Ordenador

Se utilizó durante todo el experimento un ordenador Dell Inspiron con procesador Intel Core i7-2670QM CPU 2.20GHz, 8 gb de memoria RAM. Este ordenador cuenta con sistema operativo Debian GNU/Linux 8 (jessie) 64-bit.

### B.1.2. Librerías y Plataformas de Desarrollo

La aplicación desarrollada utiliza como plataforma principal **Python** en su versión 2.7.9 y un conjunto de librerías y frameworks de procesamiento de texto los cuales se detallan a continuación:

- **NLTK** [1]: Natural Language Toolkit es una librería de Python que permite realizar análisis y procesamiento linguístico. En este trabajo utilizamos NLTK principalmente para tareas de pre-procesamiento de texto.
- **Scikit-learn** [2]: Es una librería de procesamiento y clasificación Machine Leaning de Python. La utilizamos para trabajar con los módulos de vectorización y clasificación de ngramas y skipgramas para generar Bag of Words (bolsa de palabras) de cada mensaje del corpus.
- **Numpy** [3]: Es una librería Python de uso científico que permite utilizar matrices y arreglos de gran tamaño. Es utilizada por Scikit-learn.

[1] http://www.nltk.org
[2] http://scikit-learn.org
[3] http://www.numpy.org

- **Weka** [4]: Framework de Machine Learning y Minería de Datos de plataforma Java. Es utilizado para realizar la clasificación automática y evaluación del modelo de detección de ironía presentado en este trabajo. Weka Explorer, la aplicación utilizada en este trabajo, utiliza un formato llamado **.arff** el cual mantiene una lista de instancias conformada por set de atributos. Para conformar un archivo **.arff** se debe especificar la naturaleza de los atributos (tipo de dato, nombre), nombre de la relación y los datos, que corresponden a valores de la naturaleza de los atributos separados por coma.
- **TreeTagger** [5]: Aplicación y librería Python que permite extraer las partes de una palabra POS(Part of Speech) y etiquetarlas.

### B.1.3. Base de Datos

Para mantener los tweets que forman parte del corpus de evaluación se tiene una base de datos MySQL [6] con la siguiente estructura:

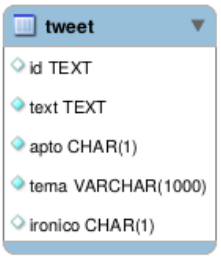


Figura B.1: Base de datos para manipulación del Corpus.

Por lo tanto, un tweet tiene un id único de identificación, el texto que corresponde al mensaje, un atributo "apto" utilizado para el filtrado manual de tweets (descrito anteriormente en el capítulo **Diseño y Construcción del Corpus de Evaluación**), "tema" corresponde a la temática relevante de bajo la cual se obtuvo el tweet y un atributo "irónico" que determina si un tweet es o no irónico.

## B.2. Arquitectura de la Aplicación

La aplicación se divide en 2 módulos, Módulo de Procesamiento y Módulo de Clasificación. El módulo de procesamiento está codificado utilizando Python y se encarga de las tareas de pre-procesado y procesado de tweets, entregando como producto un archivo dataset **.arff** que puede ser procesado por Weka. El segundo módulo es una aplicación de Weka la cual permite cargar el archivo dataset **.arff** y realizar las tareas de clasificación. El siguiente diagrama resume la arquitectura del Módulo de Procesamiento.

[4] http://www.cs.waikato.ac.nz/ml/weka
[5] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger
[6] https://www.mysql.com

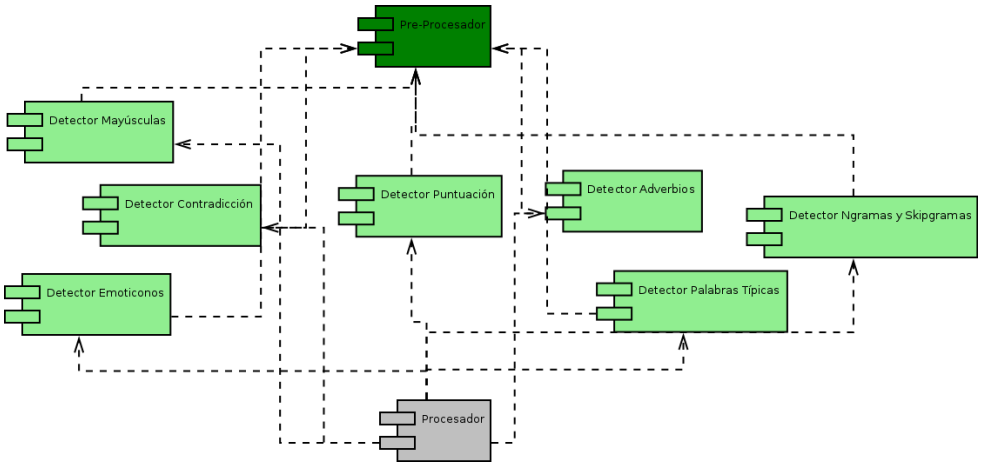


Figura B.2: Arquitectura de Procesamiento de Características.

El componente **Procesador** es un script Python que incluye a los demás componentes (clases Python) los cuales se encargan de procesar cada característica por separado. Todos estos "procesadores" a su vez usan el componente **Pre-Procesador** que es el encargado de la limpieza del texto, eliminación de caracteres especiales, etc. Este procedimiento se detalla más adelante. Finalmente, el procesador entrega un archivo **.arff** el cual corresponde al dataset que consume la aplicación Weka para la clasificación.

## B.3. Implementación del Pre-procesamiento del corpus

Se describen a continuación las funciones utilizadas para cada fase de pre-procesamiento del modelo de detección de ironía.

- **Tokenización:** Esta etapa consiste en descomponer un elemento del corpus (tweet) en distintas partes, utilizando el método **tokenize()** de la clase **TweetTokenizer()** provísta por la librería NLTK **tokenize_twitter()**. Esta función permite mantener en un arreglo cada parte del tweet (palabras, signos de puntuación, hipervínculos, hashtags, etc.) para un posterior análisis. Además, en casos puntuales se utiliza otra función de la librería NLTK **regexp_tokenize()** que permite tokenizar mediante una expresión regular.
- **Filtrado de Emoticonos:** Para filtrar emoticonos se cuenta con un diccionario de elaboración propia con la mayoría de los emoticonos de redes sociales (específicamente Twitter). Una vez tokenizado un elemento del corpus, se procede a consultar con el diccionario de emoticonos para determinar cuál o cuales contiene.
- **Filtrado de Mayúsculas:** Para obtener las palabras que son mayúsculas dentro de un elemento del corpus se utiliza la función Python **is_supper()** que entrega true si una cadena de caracteres está compuesta por letras mayúsculas.
- **Filtrado de caracteres no alfabéticos:** Mediante las funciones de Python **is_alpha()**, **is_digit()** es posible filtrar cada token que no sea una palabra (emoticonos, hipervínculos, hashtags, etc) y entregar una versión sanitizada de cada elemento del corpus.

- **POS-Tag:** Utilizando la librería Python **TreeTagger** extraemos cada Parte de la Palabra (Part of Speech) y su etiqueta correspondiente.
- **Transformar sólo minúsculas:** Algunas de las características del modelo de detección de ironía exigen que las todas las palabras del corpus estén en minúsculas. Para lograr lo anterior, se utiliza la función Python **lower()** que entrega una versión de una cadena de caracteres en letras minúsculas.