Instituto de Computación, Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

PROYECTO DE GRADO INGENIERÍA EN COMPUTACIÓN

Determinación de la orientación semántica de las opiniones transmitidas en textos de prensa

Guillermo Dufort y Álvarez
Fabián Kremer
Gabriel Mordecki

Tutor del proyecto:
Aiala Rosá, Universidad de la República

15 de julio de 2016
Montevideo, Uruguay

Determinación de la orientación semántica de las opiniones transmitidas en textos de prensa

RESUMEN

El análisis de sentimiento es una de las áreas del Procesamiento del Lenguaje Natural que más interés ha despertado en los últimos años por sus posibles aplicaciones académicas y en la industria.

Una de sus aplicaciones es determinar la orientación semántica de una opinión. En este proyecto se construye un módulo de análisis de sentimiento para integrarse con BuscOpiniones: un sistema que recupera opiniones presentes en artículos de prensa uruguaya y permite realizar búsquedas sobre ellas.

Con ese objetivo se crean tres clasificadores de sentimiento: uno basado en reglas, otro en aprendizaje automático y el restante con un enfoque híbrido cuyo objetivo es aprovechar las ventajas de los dos anteriores. Además, para poder medir la efectividad de los clasificadores, se crea un corpus de opiniones extraídas por BuscOpiniones.

El clasificador híbrido es el que obtiene los mejores resultados, mejorando de forma significativa los obtenidos por el algoritmo propuesto como línea base.

Palabras clave: análisis de sentimiento, aprendizaje automático, métodos basados en reglas, métodos híbridos, procesamiento del lenguaje natural, corpus, prensa.

Índice

Capítulo 1

Introducción

El análisis de sentimiento es una de las áreas del Procesamiento del Lenguaje Natural que más interés ha despertado en los últimos años. Sus posibles aplicaciones pueden ser tan útiles como diversas, por lo que es un campo atractivo tanto para la academia como para la industria.

Por otro lado el crecimiento de la web ha permitido la proliferación de opiniones públicas y accesibles en artículos de prensa, blogs, redes sociales, foros y otros formatos. La gran cantidad de material opinado de fácil y libre acceso ha hecho imposible para los humanos abarcar la información disponible, pero a su vez ha beneficiado la construcción y el perfeccionamiento de métodos automáticos de análisis de opiniones que facilitan esta tarea.

Dentro de este contexto, el Grupo de Procesamiento de Lenguaje Natural del Instituto de Computación de la Facultad de Ingeniería desarrolló un programa llamado BuscOpiniones que detecta y recolecta opiniones en medios de prensa uruguayos y permite la búsqueda de las mismas.

El proyecto de grado que se describe en este informe se trata de una extensión de BuscOpiniones mediante la incorporación de un módulo que permita decidir automáticamente si esas opiniones son positivas, neutrales o negativas. Es decir, incorporar el análisis de sentimiento.

El campo del análisis de sentimiento, pese a que ha experimentado un importante crecimiento en los últimos años debido a sus variadas aplicaciones y el interés tanto académico como privado, se mantiene como un problema desafiante y aún en pleno desarrollo.

Existen varios tipos de tareas a resolver, ya sea por los distintos textos de entrada o la salida esperada. Se puede realizar análisis de sentimiento a nivel de documento, como en una crítica de una película; de oración o de característica, en el caso de que en un mismo texto se opine sobre más de un objeto o más de una de las partes que lo componen.

Por otra parte, la clasificación se puede realizar en polaridad de sentimiento con dos clases positivo y negativo, se puede agregar la categoría neutral, o se pueden generar distintas escalas. Además, se puede realizar en otra gama de sentimientos como la clasificación en los sentimientos primarios: amor, alegría, sorpresa, enojo, tristeza y miedo.

En este proyecto, se tomará como entrada del algoritmo de análisis de sentimiento una opinión con su texto, fuente, y fecha como la siguiente, y se construirá una salida que indique si es positiva, neutral o negativa.

> *«¡Disfruten, vecinos, disfruten! Se ve precioso desde acá, y estoy segura de que será un antes y un después para la ciudad, para el transporte colectivo y para este barrio» dijo sobre un estrado en la terminal Colón, que es parte de la obra*
>
> Ana Olivera (1.1)

Para la opinión en 1.1, por ejemplo, la salida del programa será «Positivo», porque quien opina lo está haciendo favorablemente, demostrando una actitud positiva hacia el tema.

Existen varios procedimientos para generar este resultado, desde simples heurísticas como contar las palabras que están en una lista de palabras positivas y una de negativas y retornar la categoría que tenga más (o neutral si es empate); hasta complejos métodos de aprendizaje automático.

Para lograr una correcta clasificación, se debe lidiar con dificultades que van desde las ambigüedades propias del lenguaje natural hasta las fuertes dependencias del contexto. Estas dificultades generan incluso que en varias ocasiones dos humanos no se pongan de acuerdo en la clasificación, ya sea por diferencias en la interpretación de la opinión como por distintas visiones u opiniones personales sobre el tema en cuestión, como puede suceder con la opinión 1.2.

> *Según Lacalle Pou, con el mismo «ímpetu» que el gobierno combate el tabaco, mantiene «tolerancia» hacia la marihuana.*
>
> Lacalle Pou (1.2)

En este caso, quienes piensan que la marihuana es peor para la salud que el tabaco pueden tomar la opinión como negativa sobre lo que hace el gobierno, como seguramente lo hizo Lacalle Pou. Sin embargo, abstrayéndose del contexto de quién habla, quienes piensan que el tabaco es malo para la salud y no así la marihuana, podrían tomarlo como un elogio hacia el gobierno. Incluso quienes no tienen un juicio sobre marihuana ni tabaco, podrían argumentar

que Lacalle Pou está simplemente comparando las dos políticas, sin emitir un juicio, y por lo tanto calificarla de neutral.

Este tipo de diferencias hacen del análisis de sentimiento un problema difícil de resolver, incluso para los humanos.

1.1. Motivación

La gran cantidad de artículos, noticias, columnas, publicaciones en redes sociales y textos en general disponibles en la red son de enorme valor para la generación de los algoritmos de análisis del lenguaje natural y son a su vez el principal motivo de su existencia, ya que su análisis e interpretación se hace inabarcable para los humanos. El procesarlos automáticamente genera la posibilidad de generar datos, resúmenes y análisis de utilidad para su entendimiento. Por ejemplo, se puede calcular el porcentaje de la polaridad de las opiniones sobre un determinado tema para tener una idea de lo que se opina en los mismos; comparar las opiniones sobre un tema según el partido político al que pertenecen las fuentes o el medio que las publicó para comprobar si efectivamente existen diferencias entre unos y otros; analizar la evolución en el tiempo de la opinión de una persona sobre un tema y un largo etcétera de posibles usos.

Por otro lado, la mayor parte de los recursos y el trabajo hecho en procesamiento de lenguaje natural, y en particular en este área, están enfocados en otros idiomas que el español, mayoritariamente en inglés. Por ende, tanto la generación de recursos como la adaptación de los métodos utilizados son fundamentales para el correcto funcionamiento de los métodos aplicados al español.

Además, por la fuerte dependencia de contexto del problema, incluso trabajos realizados para el español pero de otros lugares pueden no tener la suficiente especificidad. Los factores que generan particularidades van desde la utilización de distintas expresiones y vocabulario típico de Uruguay, como el verbo *chambonear* y la expresión *ni fu ni fa* en la opinión 1.3, hasta el contexto socio-político del país. Estas diferencias hacen que la mejor forma de obtener buenos resultados sea con un programa hecho especialmente para opiniones uruguayas.

> *«Me parece que en el manejo financiero la chamboneamos», expresó y aseguró que «hay cosas que se decidieron que el directorio ni fu ni fa»* (1.3)
>
> José Mujica

1.2. Objetivos

En esta sección se explican los principales objetivos planteados en el proyecto.

Objetivos generales

El proyecto que se describe en este informe tiene los objetivos de investigación e implementación de un sistema de análisis de sentimiento, así como la integración con el ya existente sistema BuscOpiniones.

Se busca estudiar el problema del análisis de sentimiento, su área de investigación y las soluciones ya propuestas así como las posibles adaptaciones específicas para este tipo de datos en particular.

Objetivos específicos

Además, se tiene el propósito de crear de un programa que implemente una solución utilizando el conocimiento adquirido y logre determinar la orientación semántica de las opiniones recuperadas con un nivel de acierto cercano al logrado en el estado del arte del área. Es decir, se busca crear un sistema que dada una opinión determine si es positiva, neutral o negativa.

Para esto, se tiene el objetivo de la recolección, modificación y/o generación de todos los recursos léxicos necesarios, tales como diccionarios y listas de palabras.

Finalmente, se busca la integración del algoritmo generado a BuscOpiniones, de forma que se pueda visualizar en el programa el resultado del análisis así como incluir el sentimiento en las posibles claves de búsqueda.

1.3. Estructura del documento

El resto del documento se estructura del modo que se describe a continuación. En el Capítulo 2 se estudia el análisis de sentimiento y se presenta una reseña de los principales trabajos relacionados al desafío abordado. En el Capítulo 3 se describe y analiza el problema a resolver, exponiendo las dificultades que supone la construcción de un clasificador de sentimiento. Luego, el Capítulo 4 resume la idea general de la implementación de la solución. Una descripción del corpus generado para el entrenamiento y test de los clasificadores de sentimiento se presenta en el Capítulo 5. Además, se estudia la concordancia entre los anotadores, determinando un límite superior de resultados posibles. En el Capítulo 6 se describe en detalle la solución llevada a cabo para la resolución del problema. Se muestran las herramientas utilizadas, las líneas base elegidas y se presenta la descripción de los tres clasificadores de sentimiento generados. En el Capítulo 7 se describe la evaluación experimental realizada sobre los clasificadores

implementados y se discuten los resultados alcanzados. En el Capítulo 8 se describe la creación e integración de un módulo de análisis de sentimiento en la plataforma *BuscOpiniones*, que utiliza el clasificador de sentimiento desarrollado que obtiene los mejores resultados. Por último, las conclusiones del proyecto y las principales líneas de trabajo futuro se presentan en el Capítulo 9.

Capítulo 2

Marco de Trabajo

En este capítulo se describen los principales trabajos relacionados al desafío abordado en este proyecto. Se analiza lo ya existente y se fija un punto de partida para la tarea a realizar.

Primero se define el análisis de sentimiento, analizando los conceptos sobre los que se construye. Luego, se describen las dificultades inherentes al análisis del lenguaje natural a la hora de determinar la polaridad de una oración, mostrando ejemplos que ilustran los distintos problemas.

Por último, se analizan las diferentes estrategias que pueden ser tomadas para implementar un clasificador, y se presentan trabajos relacionados a la tarea a realizar.

2.1. Análisis de sentimiento

El análisis de sentimiento es una rama del Procesamiento de Lenguaje Natural que estudia la identificación y extracción de la información subjetiva de un texto.

La información textual existente en el mundo, en general, puede ser categorizada en dos principales categorías: *hechos* y *opiniones*. Los hechos son expresiones objetivas sobre entidades, eventos y sus propiedades. Las opiniones son usualmente expresiones subjetivas que describen los sentimientos de personas, apreciaciones o sensaciones hacia entidades, eventos y sus propiedades [16]. A partir de esa división planteada se desprende naturalmente la cuestión de qué elementos del texto son los que distinguen un hecho de una opinión.

2.1.1. Diferencia entre hecho y opinión

Los autores de [33] proponen un sistema de anotación de opiniones, donde se introduce el concepto de *estado privado*. Un *estado privado* es un término general que abarca opiniones, creencias, pensamientos, sentimientos, emociones, objetivos, evaluaciones y juicios. Este esquema es explicado en profundidad en el apéndice .

En [24] se define a un *estado privado* como un estado que no puede ser observado o verificado de forma objetiva:

> *«Es posible observar a una persona afirmando que Dios existe, pero no creyendo que Dios existe.»* [1] Quirk et al. (2.1)

Los estados privados se pueden ver como individuos experimentando actitudes, en ocasiones dirigidas hacia objetivos; y se detallan distintas propiedades que pueden asociárseles tales como: un *fragmento de texto* que expresa el estado, un *experimentador*, un *tipo de actitud* y una *intensidad*. Por ejemplo, para el estado privado manifestado en la oración *"Juan odia a María"*, el texto que expresa el estado privado es la palabra *odio*, el experimentador es *Juan*, la actitud es *odio*, el objetivo es *María* y la intensidad *alta*. Puede suceder entonces que la diferencia entre una oración objetiva y una opinión sea la existencia de estados privados expresados en las mismas, que se identifican por fragmentos de texto que expresan una actitud de una fuente hacia un objeto.

2.1.2. Clasificación de polaridad

Un aspecto importante a considerar es qué tipo de actitud puede atribuirse a un estado privado. Existen diversos estudios en lingüística, psicología y análisis de contenido en tipología de actitudes ([13–15, 31]) que proponen clasificaciones de actitudes en distintos subtipos como *emoción*, *advertencia*, *postura*, *incertidumbre*, *condición*, *cognición*, *intención*, y *evaluación*. A pesar de que los estados privados pueden ser clasificados en muchos subtipos de actitud, la mayoría de los trabajos relacionados a la clasificación de sentimientos reduce las actitudes a una clasificación binaria entre positivo y negativo, o a ubicar la actitud en un rango entre las dos opciones que pueden ser n opciones discretas o un continuo entre ellas. A la tarea de etiquetar un documento u oración subjetiva, con una clasificación binaria, donde la fuente expresa en general una opinión positiva o negativa, se le llama clasificación de polaridad sentimental, o clasificación de polaridad [16].

[1] Traducción de los autores del original en inglés: «a person may be observed to assert that God exists, but not to believe that God exists».

El proceso que se lleva a cabo para determinar la polaridad sentimental de un elemento depende de si es un documento, oración, expresión o palabra. A medida que el elemento de estudio es más complejo, y de mayor cantidad de palabras, la percepción de la polaridad del mismo puede variar. Por ejemplo, la polaridad de una palabra sin contexto puede ser determinada a través del análisis de su significado, mientras que la de una oración es el sentimiento general expresado por la misma. A nivel de documento, a su vez, puede suceder que la mayoría de las oraciones tenga una polaridad positiva pero que la polaridad final sea negativa si la conclusión también lo es.

En este proyecto el objeto de estudio son opiniones extraídas de la prensa, que en general constan de una o más oraciones, pero que no alcanzan a tener la extensión de un documento. A partir de esto, la mayoría de este trabajo se basa en métodos y trabajos que se enfocan en determinar la polaridad de sentimiento a nivel de oración.

2.1.3. Análisis a nivel de oración

Los estados privados son expresados a través de fragmentos de texto. Estos fragmentos de texto son en su estado más básico términos simples, palabras, a los que se le atribuye una polaridad sentimental base. La polaridad sentimental base de una palabra, debe ser determinada a partir del significado de la palabra y responder la pregunta: fuera de contexto, ¿la palabra parece evocar algo positivo, negativo o neutral? Ejemplos de palabras cuya polaridad es positiva son: *aprobación*, *hermoso*, *bueno*, etc. Mientras que palabras cuya polaridad es negativa son: *desastre*, *horrible*, *insulto*, etc.

Para determinar la polaridad de una oración, algunos investigadores han recurrido a métodos de simples conteos de términos positivos y negativos ([12, 30, 32]). Estos métodos pueden dar resultado, como en la oración 2.2, donde existen dos palabras de polaridad sentimental positiva y la polaridad total de la oración también lo es. Por otro lado el conteo de palabras no siempre es efectivo ya que no tiene en cuenta la relación entre las mismas, ni toma información del contexto lingüístico en el que son utilizadas. En el ejemplo 2.3 la oración es negativa por más que la suma de términos es positiva, ya que el *no* tiene un efecto de inversión sobre la polaridad de los términos sobre los que tiene alcance.

A Juan le ***encantan***$^{+}$ *las manzanas y las* ***disfruta***$^{+}$ *siempre por la mañana.* (2.2)

A María ***no*** *le* ***gusta***$^{+}$ *correr por el parque.* (2.3)

Los negadores e intensificadores, la ironía y los elementos presuposicionales, entre otros, son elementos que influyen sobre la polaridad de una oración, y dificultan la tarea de determinar

la polaridad resultante de la misma a partir de sus términos. El detalle, junto con ejemplos, de los elementos lingüísticos que cambian las polaridades de los términos, son presentados en la siguiente sección.

2.2. Polaridad contextual

Determinar la polaridad de una oración a partir de la polaridad de términos individuales puede llevar a resultados equivocados, como fue explicado en la sección previa. Esto se debe a la presencia de elementos léxicos que pueden modificar la polaridad de las palabras con las que interactúan, pudiendo invertir la polaridad, o llevarla hacia la neutralidad. Estos elementos léxicos son estudiados en [23], donde se los denomina *elementos que cambian la polaridad*. Además, los autores detallan una división de los elementos en distintas clases que se presenta a continuación:

- **Negadores:** Los negadores son elementos que invierten la polaridad de las palabras sobre las que tienen alcance. Cómo modelar el efecto de la negación es un problema que ha sido estudiado ampliamente en la literatura ([9, 22]). Ejemplos de negadores simples incluyen a: *no, nunca, ningún, nadie, nada, tampoco, etc.* Por otro lado existen negadores más complejos, denominados *negadores con contenido*, que también pueden invertir la polaridad de otras palabras como: *terminar, destruir, atacar*, etc.

- **Intensificadores:** Otro tipo de modificadores de polaridad son los intensificadores, que, a diferencia de los negadores, no invierten la polaridad del elemento que afectan, sino que fortalecen o debilitan su intensidad. Ejemplos de intensificadores son: *muy, demasiado, casi, gran, un poco*, etc.

- **Operadores modales:** En el lenguaje se hace distinción entre eventos que se confirma que ocurrieron, que están ocurriendo, o que van a suceder (eventos *realis*) y los que podrían, deberían, o posiblemente ocurrieron o van a ocurrir (eventos *irrealis*). Por ejemplo, en la opinión *"María es una persona* ***horrible****.Ella trata* ***mal*** *a los perros."*, **horrible** y **mal** son elementos de polaridad negativa y como resultado ambas opiniones son negativas. Sin embargo, la oración *"****Si*** *María fuera una persona horrible,* ***trataría*** *mal a los perros."* no está diciendo ni que María es horrible ni que trata mal a los perros. Al contrario, la fuerza del condicional del **trataría** sugiere que *ella no es mala hacia los perros* mientras que el **si** determina un contexto en el cual María no es necesariamente una mala persona. De manera que la presencia de estos términos neutraliza las oraciones.

- **Elementos presuposicionales:** Existen palabras que cambian la polaridad de términos evaluativos mediante sus presuposiciones. Esto es típico de adverbios como *apenas*

como puede ser visto comparando las expresiones *"Es suficiente"* con *"Es apenas suficiente"*. *Suficiente* es un término positivo, pero *apenas suficiente* no lo es: presupone que debería ser mejor. Este tipo de términos pueden introducir una evaluación positiva o negativa aun sin la presencia de elementos evaluativos como en *"Apenas pudo ingresar a la Universidad"*.

- **Ironía:** Existen casos en los que aún tomando en cuenta los elementos descritos anteriormente se puede interpretar erróneamente la polaridad de una opinión. Por ejemplo, en la opinión *"El tan excelente organizador falló en resolver el problema"* la connotación extremadamente positiva de *tan excelente* se invierte ya que el autor lo está diciendo irónicamente.

- **Conectores:** Los conectores como *aunque, pero, sin embargo, al contrario, etc*, pueden tanto introducir información, o actuar sobre información en otra parte del texto para mitigar su fuerza. Por ejemplo en la oración *"Aunque Juan es **excelente** en matemáticas, es un **mal** profesor"*, por más que *Juan es **excelente*** tiene una polaridad sentimental positiva, la fuerza del *aunque* combinada con una expresión de polaridad negativa posterior niega el efecto de lo positivo.

- **Discurso indirecto:** Si consideramos la oración *"Juan es **tonto**"*, la polaridad contextual de la oración es negativa debido a que la polaridad base de *tonto* es negativa y no existe otro elemento que cambie la polaridad que lo afecte; por lo tanto podemos decir que el autor opina negativamente sobre Juan, ya que dice que es tonto. Por otro lado, si tomamos la oración *"María dijo que Juan es **tonto**"*, el autor afirma que *María* dijo algo negativo sobre *Juan*; lo que no quiere decir que el autor acepte lo que María dice. Sin embargo, información que se encuentre luego en el texto puede de todas formas forzar su inclusión, como por ejemplo: *"María dice que Juan es tonto, **y tiene razón**"*.

- **Restricciones de contexto:** Al considerar la polaridad sentimental léxica de un término sin contexto como positiva o negativa se está realizando una simplificación. Existen casos donde, dado un contexto específico, un término puede ser positivo, pero en otros contextos puede ser negativo; como por ejemplo **rápido** en *"El corredor es muy **rápido**"* es claramente positivo. Sin embargo en *"la batería se acaba muy **rápido**"* tiene una connotación negativa clara. Otros ejemplos de esta categoría son:*grande, chico, lento, barato, etc.*

En esta sección se presentaron distintos elementos que ocurren en el texto e influyen sobre la polaridad sentimental del mismo. La detección de ocurrencias de estos elementos y el alcance e influencia que tienen son una tarea compleja que ha sido ampliamente estudiada en la literatura. Los enfoques más utilizados para la clasificación sentimental de texto son analizados a continuación.

2.3. Métodos de clasificación

En la literatura se han estudiado diferentes enfoques a la hora de crear sistemas para determinar automáticamente la polaridad de un fragmento de texto. La mayor parte de los enfoques pueden categorizarse en tres grandes clases: los basados en reglas, los que utilizan aprendizaje automático, y los enfoques híbridos que combinan los dos anteriores [5]. Los tres tipos de enfoque comparten el mismo objetivo: asignar a un fragmento de texto, de forma automática, una etiqueta extraída de un conjunto finito especificado con anterioridad, mediante la aplicación de un algoritmo de clasificación.

2.3.1. Métodos basados en reglas

Los métodos basados en reglas tradicionalmente se basan en la utilización de lexicones de sentimiento: conjuntos de listas de palabras clasificadas con un valor de sentimiento. A partir de estos, se construye un sistema, que utilizando heurísticas lingüísticas como conteo de palabras positivas y negativas, calcula la polaridad sentimental resultante para el texto de entrada. Métodos más complejos hacen uso de la información sintáctica y semántica del texto introduciendo en las heurísticas elementos como negadores e intensificadores que influyen en la polaridad total de la opinión.

La validez de los métodos basados en reglas depende en gran medida de la profundidad y amplitud de los recursos empleados. Sin una amplia base de conocimientos que abarque el conocimiento humano no es fácil para un sistema de análisis de sentimiento captar la semántica asociada con el lenguaje natural o el comportamiento humano.

2.3.2. Métodos basados en aprendizaje automático

El Aprendizaje Automático es una rama de la Inteligencia Artificial que desarrolla programas capaces de aprender a resolver problemas mediante algoritmos de aprendizaje y generación de conocimiento. En [18] se define formalmente de la siguiente forma:

> *Un programa aprende de la* ***experiencia E*** *a realizar una determinada* ***tarea T*** *si su desempeño en la tarea T, medida con la* ***métrica M****, mejora con la experiencia E.* [2] [Machine Learning. Thomas M Mitchell, 1997] (2.4)

[2] Traducción de los autores del original en inglés: «A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E».

Por ejemplo, un programa puede aprender a jugar al Go [3], en este caso sería la tarea T. Lo puede hacer mirando ejemplos de partidos jugados anteriormente (la experiencia E) y con la métrica de la cantidad de partidos ganados ponderado por la dificultad de los oponentes humanos (métrica M).

Para que el algoritmo pueda aprender, se le da ejemplos de entrenamiento que cumplen el rol de la experiencia E. Esos ejemplos son representados con un conjunto fijo de atributos (llamados ***features***). Estos atributos forman un vector, normalmente numérico, que describe los ejemplos con información que pueda ser relevante para el algoritmo.

Existen varias ramas del aprendizaje automático que se dedican a lidiar con diferentes tipos de tareas de aprendizaje. Estas ramas pueden ser distinguidas con varios criterios tales como *activo* o *pasivo*, *en directo* o *en batch* o una de las más comunes: *supervisado* o *no supervisado*.

Se dice que el aprendizaje es **supervisado** cuando la *experiencia* de la que aprende el algoritmo contiene información relevante que no aparecerá en los ejemplos de prueba, como una etiqueta para cada ejemplo que señala la salida esperada ante ese ejemplo. Es **no supervisado** si los ejemplos de entrenamiento son iguales que los que se usarán luego. [27]

Uno de los problemas más comunes a resolver en el campo del aprendizaje automático es el de la clasificación: asignar categorías o etiquetas de un conjunto discreto y finito a elementos de un dominio dado.

El análisis de sentimiento, en particular a nivel de oración, se modela normalmente como un problema de clasificación supervisado [17].

Como este tipo de métodos está basado en el comportamiento estadístico de las muestras de entrenamiento, depende fuertemente del tamaño y correctitud del conjunto utilizado para entrenar que, mientras más ejemplos tenga, más información va a poder extraer el algoritmo. Generalmente los clasificadores entrenados mediante aprendizaje supervisado son buenos aprendiendo la polaridad sentimental de las palabras, así como la polaridad conjunta de co-ocurrencias de términos. Sin embargo, les es difícil aprender las relaciones semánticas entre las palabras, por lo que usualmente son más efectivos para clasificar texto a nivel de párrafo o página, pero suelen tener problemas a la hora de analizar unidades pequeñas de texto como oraciones o frases [7].

2.3.3. Métodos híbridos

Los métodos híbridos, como su nombre indica, son combinaciones de los dos métodos descritos anteriormente. Tanto los métodos basados en reglas, como los que utilizan aprendizaje tienen sus ventajas y desventajas, y requieren de recursos y herramientas para su realización. Este tipo

de métodos investigan cómo llevar la información del análisis basado en reglas a los algoritmos de aprendizaje, o viceversa, de manera de aprovechar lo mejor de ambos métodos, mitigando las posibles limitaciones de los recursos utilizados.

2.4. Trabajos relacionados

Los enfoques mencionados en la sección anterior son reiteradamente utilizados en la literatura del análisis de sentimiento, pero la mayoría son para textos en inglés. Es importante resaltar que aunque existen similitudes entre las funciones sintácticas y semánticas que cumplen los elementos de los distintos lenguajes, las estructuras sintácticas, así como los léxicos, suelen ser diferentes. Esto hace que se torne compleja la reutilización de herramientas construidas para el inglés (lexicones, datos de entrenamiento, etc) para analizar textos en español, que es el idioma que se utiliza en este trabajo.

Además, los distintos dominios del lenguaje sobre los que se aplican los algoritmos, así como los tamaños de los textos, tienen una gran influencia sobre los resultados. En consecuencia, comparar en valor la efectividad de algoritmos probados sobre diferentes conjuntos de datos de test, y sobre todo si son en distintos lenguajes, suele llevar a conclusiones erróneas.

Teniendo en cuenta los puntos mencionados, se realiza una reseña de algunos trabajos relacionados al análisis de sentimiento, para dar un panorama general del estado del arte, y ofrecer una idea general de las estrategias utilizadas para abarcar el problema.

En [29] se presenta un clasificador que contempla muchos de los elementos tratados en [23], en su algoritmo de cálculo basado en reglas. Algunos de los elementos que contempla son: negadores, intensificadores y operadores modales.

Un aspecto interesante del algoritmo, es que se plantea que el lenguaje tiene una tendencia a ser positivo, por lo que se pondera con mayor peso a las palabras negativas (se les asigna un 50 % más de peso). Otro punto que plantea es que la repetición de palabras influye en que por cada repetición de la misma en el fragmento de texto, su peso disminuye debido a que es posible que se trate del tema del texto. De esta manera la influencia de la palabra en la opinión muestra una tendencia hacia la neutralidad.

Un punto interesante en el algoritmo que determina la influencia de la negación es que no se utiliza la negación como un *inversor de polaridad*. Si se toma la opinión *"Esta película es buena"* se puede decir que es una opinión positiva respecto a la película, mientras que la misma opinión pero negada *"Esta película no es buena"* es negativa, por lo que la inversión parece lógica. Sin embargo la opinión positiva negada *"Esta película no es excelente"* tiene una tendencia a la neutralidad más que la negatividad. La forma en que se propone resolver

este problema es asignando a cada palabra un valor de polaridad (reflejado en el signo del valor atribuido) y una intensidad (reflejado en la magnitud del valor). Luego la negación es un corrimiento fijo en sentido contrario a la polaridad de la palabra. Si *"buena"* es +2, y la negación es un corrimiento de magnitud 4, *"no buena"* pasa a ser -2 como resultado, mientras que si *"excelente"* es 4, *"no excelente"* es 0, lo que se acerca más a la noción intuitiva del lenguaje. Los resultados presentados indican que en términos de *accuracy* se mejora a la línea base de conteo de palabras de un 66,04 % a un 78,74 % de accuracy utilizando todas las features mencionadas. Esto es una mejora de un 14,70 % en total, teniendo en cuenta que es clasificación binaria entre positivo y negativo.

En otro trabajo [19], también se intenta integrar la relación entre las palabras al algoritmo de clasificación, pero lo que se pretende es aún más ambicioso. Los autores proponen que es posible calcular de manera sistemática la polaridad de constituyentes sintácticos grandes, como función de sus sub-constituyentes, de manera casi análoga al principio de composicionalidad. Si el significado de una oración es una función del significado de sus partes, entonces la polaridad global de una oración es una función de la polaridad de sus partes.

Con esta premisa como base se define un modelo composicional formado por reglas básicas que toman como entrada dos constituyentes sintácticos y calculan una polaridad global para el constituyente de salida. A partir del modelo definido se construye un clasificador que toma como entrada lexicones de sentimientos de palabras, y la salida de analizar la oración con un parser de dependencias.

Luego aplican las reglas definidas en el modelo teórico desde el constituyente de la oración ubicado más hacia la derecha, hasta llegar a un criterio de parada. El método propuesto intenta capturar la información sintáctica de la oración para poder determinar el alcance y la influencia de los distintos fenómenos lingüísticos que alteran la polaridad global de la oración.

Además de los resultados obtenidos se presenta un análisis sobre los mismos, que arroja conclusiones muy interesantes. Como se encuentra planteada la solución, el método depende en su totalidad de que el parsing de dependencias, junto con el tagging de las palabras, funcionen correctamente. En el análisis se reporta que un 28 % de los errores que se detectaron ocurrieron a causa de errores propagados de un mal funcionamiento de las herramientas que utilizaron. Otro aspecto que se destaca es que en el análisis se concluye que aún teniendo una entrada correcta, el modelo propuesto igual no podría resolver un 19 % de los casos en los que se dan errores debido a la falta de conocimiento del contexto.

En [33] se intenta a partir de palabras (llamadas *clues* en el trabajo) y sus respectivas polaridades a priori, determinar las polaridades con las que actúan en distintos contextos.

El primer paso que se realiza es clasificar la expresión que contiene a la palabra entre polar y neutro. El segundo paso toma todas las expresiones marcadas como polares y desambigua

su polaridad entre positiva, negativa, ambas y neutral. Para lograr diferenciar expresiones neutrales de polares los autores proponen realizar un clasificador con una serie de features de distintos tipos:

- **de palabra:** cuáles son las palabras adyacentes a la *clue*.
- **de modificación:** qué relación sintáctica tienen las palabras adyacentes a la *clue*.
- **de estructura:** a qué relaciones sintácticas dentro de la oración pertenece la *clue*.
- **de oración:** contadores de *clues* en la oración actual, la anterior y la posterior.
- **de documento:** una feature indicando el tema del documento.

Para el clasificador entre neutral y polar lograron un accuracy del 75,9 %, que mejora al clasificador de línea base que tiene como única feature la *clue* en un 2,3 %. Para el clasificador que desambigua la polaridad se utilizaron features que intentan capturar la influencia de la negación (búsqueda de negadores cercanos, o que sean hijos en el árbol de dependencias), y features que representan si la clue es modificada por otras palabras polares cercanas que sean adjetivales, o se encuentren en conjunción.

Con el segundo clasificador logran clasificar entre 4 clases distintas (positiva, negativa, ambas y neutral) con una accuracy de 65,7 % (la segunda clasificación se realiza sobre instancias ya clasificadas como polares), mejorando al clasificador línea base que utiliza como única feature la clue en un 4 %.

En [8] también se utilizan los conceptos de semántica composicional, y se da un paso más al integrarlos en un algoritmo de aprendizaje automático supervisado. Para lograrlo, en principio se utiliza un conjunto de reglas de composición determinadas a mano que funciona de forma similar a las reglas presentadas en [19].

La diferencia principal es que para aplicar las reglas se utiliza un conjunto de variables, determinadas a través de un clasificador, que reemplazan el lexicón de sentimiento y la lista de negadores. A cada palabra perteneciente a la expresión se le asigna con el clasificador una etiqueta que puede ser: positiva, negativa, negador o ninguno. Luego a partir de la oración original y los nuevos valores de cada palabra, se utilizan las reglas y se determina el resultado de la expresión.

Los resultados obtenidos indican que, a diferencia de un método simple basado en conteo de palabras positivas y negativas que arroja un accuracy de 86,5 %, con el método basado en semántica composicional basado en reglas se alcanza un accuracy de 89,7 %, mientras que con aprendizaje supervisado aumenta a 90,7 %. Se puede decir que la introducción de la semántica

composicional aplicada a un método supervisado logra una mejora respecto a la línea base de un 4,2 %, siempre teniendo en cuenta que el análisis es a nivel de expresión y su clasificación es entre dos clases: positivo y negativo.

Capítulo 3

Análisis del problema

Este capítulo presenta una descripción del problema abordado en el marco de este proyecto, que consiste en aplicar los conocimientos del área del análisis de sentimiento en la creación de un programa que permita detectar automáticamente la polaridad de una opinión, e integrarlo a la plataforma BuscOpiniones.

3.1. Estudio del corpus

Como parte de la exploración inicial y análisis del problema, se realizó un estudio del corpus existente de BuscOpiniones.

El estudio consistió en una categorización de polaridad detallada de un conjunto de opiniones junto con el análisis de los problemas y dificultades encontrados. Además, se marcaron los principales elementos que hacen al anotador percibir la subjetividad de la oración, así como la posición de su autor sobre el tema del que habla.

Se utilizó un esquema de anotación basado en el propuesto por Wiebe et al. [33] y posteriormente modificado de forma de que se adecuara mejor a lo buscado en esta etapa del proyecto.

En esta etapa se logró estudiar el problema y el corpus con el que se trabaja en este proyecto y conocerlo en profundidad.

Se encontraron problemas como las imperfecciones provenientes de BuscOpiniones para recuperar las oraciones bien formadas, las faltas de contexto y una concordancia entre anotadores menor a la esperada. Todos estos problemas encontrados serán profundizados en el resto de este capítulo.

Además, se descartó la utilización de algunos de los métodos estudiados en los trabajos relacionados, ya que se encontró que no se ajustan al problema planteado. Por ejemplo, se descartaron

los que sugieren la utilización únicamente de los adjetivos para el análisis, ya que se anotaron otros tipos de palabras como sustantivos y conectores como fundamentales para conocer la polaridad.

Finalmente, se generó un esquema que puede ser de utilidad para realizar una anotación de sentimiento detallada. Sin embargo, se observó que el esquema es demasiado profundo y verboso como para permitir una anotación rápida, por lo que no es adecuado para la etapa de anotación de un corpus.

El esquema utilizado, los cambios realizados y el detalle del proceso de estudio del del corpus se encuentran en el anexo A.

3.2. Criterios de anotación

Durante el análisis realizado en la primera fase de anotación se encontraron dificultades en reiteradas ocasiones al momento de decidir la polaridad de las opiniones. Para minimizar las diferencias y unificar los reglas según las que se decide cómo anotar, fue necesario determinar un criterio único para los problemas encontrados. En esta sección se describen algunos de ellos y los criterios tomados para cada uno.

3.2.1. Criterio para la dependencia del contexto

En una gran cantidad de ocasiones las opiniones carecen de contexto, ya sea por la falta del resto del artículo del que fue recuperada la opinión o por información externa que no se encuentran en el texto pero asumidos como de conocimiento de los lectores tanto por quien emite la opinión como por quien escribe el artículo.

Existen muchos tipos de dependencia de contexto, a continuación se describen algunas de ellas:

- **Contexto textual:**

  Para Suárez parecen no existir los límites.

  Suárez (3.1)

  En la opinión 3.1 es necesario conocer el resto del texto que acompaña la oración para saber a qué se refiere. «No existen límites» puede ser positivo cuando se habla por ejemplo de un jugador de fútbol, o negativo en otros contextos como si se habla de un criminal. Por esta razón, sin saber a qué se dedica «Suárez» o sobre qué se habla, no es posible determinar la polaridad de la opinión.

- **Contexto temporal:**

> *Días atrás, la intendenta Ana Olivera adelantó que las obras viales que se están realizando en la capital del país finalizarán en los meses que restan del año 2011.* (3.2)
>
> Ana Olivera

En casos del tipo de la opinión 3.2 es crucial conocer el momento o la coyuntura temporal en que fue emitida la opinión. En este ejemplo, debe saberse cuánto restaba del año 2011 cuando se dijo eso y cuándo se planeaba terminar la obra originalmente para saber si sería en tiempo o no. Incluso sucede que la fecha que recupera BuscOpiniones no es correcta, ya que está registrada como de octubre de 2012 y la opinión tiene que haber sido en el año 2011, por lo que aún en el caso de haber tenido en cuenta este dato hubiera generado una información errónea.

- **Contexto político:**

> *Mujica advirtió que su grupo sigue «peleando» por Rosadilla.* (3.3)
>
> Mujica

Para clasificar correctamente esta opinión, se debe saber el contexto político en el que fue dicha. En el caso de que hubiera sido debido a que Rosadilla generó una disputa interna, sería negativa. Sin embargo, Mujica hizo esta advertencia ya que estaba insistiendo en la protección de su compañero de sector, en ese momento ministro de Defensa. Sin tener esta información política, es imposible saber cuál de los significados la palabra *pelea* es la utilizada y por lo tanto la polaridad de la opinión.

- **Contexto socio-cultural:**

> *«Muchas de las cosas que hicimos frente a Venezuela le pueden doler a cualquier equipo», indicó Tabárez.* (3.4)
>
> Tabárez

Aquí, sin contexto, se puede interpretar esta oración como una amenaza más que una opinión. Sin embargo, sabemos que Tabárez es el técnico de la selección uruguaya y en el contexto deportivo es sabido que «hacer doler» es dicho en sentido figurado y además que hacer daño al equipo rival es algo positivo. El contexto cultural nos hace saber que la opinión es positiva, si no supiéramos que es deportiva, bien podríamos haberla calificado como negativa.

En el caso de la falta de contexto, se definió que las opiniones sean consideradas junto con su contexto social, cultural y político, ya que es justamente lo que se busca al realizar un proyecto con opiniones locales. Por otro lado, la expansión de contexto temporal en el texto se evaluó como fuera del alcance del proyecto, por lo que se considera únicamente la información que está presente en el texto recuperado y no la que lo acompañaba en el artículo original.

3.2.2. Criterio para múltiples polaridades

En algunos de los textos recuperados existe más de una opinión, incluso a veces con polaridades opuestas. Esto sucede por la presencia de multiplicidad de temas, variedad de juicios sobre lo mismo o una mezcla de esto, como se detalla a continuación:

- **Múltiples tópicos, única polaridad:**

  En una misma oración se opina sobre más de un asunto, o el texto contiene más de una oración y contienen distintos tópicos entre ellas. Todas las opiniones coinciden en la polaridad.

> *«Los jugadores terminaron extenuados, porque para nosotros este partido no comenzó cuando el juez norteamericano pitó el inicio, lo estábamos jugando desde que terminó el partido frente a Perú. Conseguimos lo que queríamos y mantenemos nuestras expectativas para superar la fase», dijo Tabárez* (3.5)
>
> Tabárez

  En la opinión 3.5, por ejemplo, Tabárez habla positivamente del *partido* y como su equipo lo enfrentó, pero también sobre las *espectativas para superar la fase*. Lo ideal sería que la opinión fuera recuperada tanto para una búsqueda de opiniones de «Tabárez sobre partidos de Uruguay» como de «expectativas de Tabárez». En este caso no hay dudas sobre la polaridad total de la opinión, ya que se marca la única presente.

- **Único tópico, múltiples polaridades:**

  En una misma oración se opina con polaridades distintas sobre un mismo tópico.

> *«Intentan adoptar medidas responsables, pero esas medidas no han sido todo lo rápidas que deberían», agregó el presidente* (3.6)
>
> el presidente

La opinión 3.6 es sobre las «medidas» que «intentan adoptar», el tópico está claro y es único. Sin embargo, se opina positivamente al decir que tienen la intención de ser «responsables» y negativamente al decir que son más lentas de lo debido.

- **Múltiples tópicos, múltiples polaridades:**

  Sucede comúnmente cuando los textos extraídos tienen más de una oración, aunque puede incluso pasar que en una misma oración se hable de más de un tópico.

> *Forlán señaló: «Uruguay no es una selección que tenga mucha suerte. Aquí tenemos que jugar muy bien para poder alcanzar nuestros objetivos. No somos como Brasil, por ejemplo, que juega muy bien, pero a la que también le acompaña mucho la suerte. Bueno, quiero aclarar que no me refiero a que Brasil gane por suerte. ¡No! Brasil tiene jugadores fantásticos, y una grandísima tradición futbolística»* (3.7)
>
> Forlán

  En la opinión 3.7 Forlán habla sobre la selección uruguaya de forma negativa («no tiene mucha suerte») y sobre la brasilera de forma positiva («tiene mucha suerte»). Incluso, en un análisis más fino, se podría incluir como tópicos también *la suerte de brasil*, *los jugadores de Brasil*, *el juego de Brasil* y *la tradición futbolística de Brasil*.

En todos estos casos lo ideal sería marcar más de una opinión por texto y clasificar la polaridad de cada uno de ellos. No obstante eso implicaría hacer un análisis más detallado que a nivel de oración, lo que no es el objetivo de este proyecto.

Por esa razón se debe anotar una única polaridad para toda la opinión y el criterio definido es que se tome la sensación general que deja, fijándose en la más importante de las polaridades que expresa la fuente. La importancia puede medirse ya sea porque es sobre el tema más relevante, porque es la que destaca la fuente mediante los conectores que utiliza, porque es la más intensa de las presentes, o por cualquier otra razón.

Continuando con los ejemplos y aplicando el nuevo criterio: la opinión 3.5 es positiva ya que todas las polaridades presentes lo son también. La 3.6 se anota negativa porque el *intentan* atenúa la polaridad positiva de la primera parte y el conector *pero* prioriza la polaridad negativa del final, como se explica en la sección 2.2. Finalmente la opinión 3.7 se anota positiva debido a que se considera que, pese a que las tres polaridades están presentes en ella, la positividad resalta ante las otras en la sensación general.

3.2.3. Criterio para el tema incierto

En muchos casos el tema de la opinión no está totalmente claro, ya sea porque falta contexto para saberlo, porque la fuente es ambigua sobre a qué se está refiriendo o porque el tema es en realidad un subtema del principal.

Se considera un subtema cuando se está hablando de una característica o particularidad, de algo que constituye una parte pero no el todo del tema principal.

> *Mujica expresó que la marihuana es «una droga relativamente benigna»*
>
> Mujica (3.8)

La opinión 3.8 fue recuperada con una búsqueda sobre "Droga". Sin embargo, la opinión no es sobre el tema droga en general sino sobre la marihuana, apenas una de las muchas drogas existentes. Incluso, la opinión de Mujica sobre la marihuana no es negativa, pese a que sí lo es (como se puede comprobar en las otras opiniones) sobre la droga en general.

Para estos casos, se decide tomar como tema de la opinión para su análisis el que más se ajuste al real, permitiendo los que son subtemas del que fue buscado, e incluso en los casos en los que el tema es directamente distinto. Por lo tanto en este caso se toma el tema como "marihuana" (no como "droga") y la subjetividad es positiva.

Además, en ocasiones el tema directamente no es conocido, ya que no se encuentra presente en el texto recuperado sino en su contexto.

Sin contar con la información del tema no se puede saber la polaridad de ciertos términos. Por ejemplo, el adjetivo *impredecible* puede ser positivo si se le atribuye a la trama de una película, pero en el caso de ser atribuido a la dirección de un auto es claramente negativo. Otros claros ejemplos de esta categoría son *grande* y *chico*, *rápido* y *lento*, etc.

> *Mujica señaló que ese concepto «es el abc» y que no es «ninguna novedad»*
>
> Mujica (3.9)

En la opinión 3.9 falta contexto para saber cuál es «ese concepto». Que algo sea conocido y básico puede ser positivo o negativo dependiendo del tema, por lo que este es uno de los casos donde es necesario el tema para conocer la polaridad de la opinión. Al entrar a la nota y leer el contexto, se puede ver que Mujica se refiere a la necesidad de agregar valor a las exportaciones de Uruguay.

Este caso es similar al que sucede con la falta de contexto (sección 3.2.1). Por lo tanto, se toma un criterio similar: se considera sólo la información disponible.

3.2.4. Criterio para la ironía

Algunas de las opiniones presentes en el corpus son irónicas, como la opinión 3.10.

> *«resulta pintoresco que luego de años de vociferar contra Estados Unidos, sea él precisamente quien haya evaluado esa medida», agregó Lacalle.* (3.10)
>
> Lacalle

La detección automática de la ironía es un problema complejo que implica todo un campo de investigación del Procesamiento de Lenguaje Natural. Este proyecto no abarca el problema de detectar ironía, por lo que seguramente este tipo de opiniones generen problemas para su correcta clasificación. Sin embargo, deben anotarse con la polaridad que realmente poseen, teniendo en cuenta el tono irónico.

3.2.5. Criterio para el sesgo de objetividades

Existen ciertas opiniones que podrían ser consideradas como objetivas, debido a que la fuente lista datos que pueden ser considerados objetivos. Sin embargo, el sesgo en la selección de los mismos genera una subjetividad, ya que se presentan sólo los negativos (o positivos) y se omiten intencionadamente los otros.

> *Para Lacalle Pou, las medidas que está empleando el gobierno en relación a la economía van a generar más déficit, más recesión y van a encarecer el costo de vida* (3.11)
>
> Lacalle Pou

En la oración 3.11 se ve un ejemplo: déficit, recesión y encarecimiento son tres consecuencias negativas y seguramente «las medidas» fueran a generar otras consecuencias positivas. Es un ejemplo de un sesgo negativo.

En estos casos es el sesgo de la selección lo que hace que la fuente exprese un estado privado subjetivo, y se decide considerarlos como opiniones con una polaridad marcada por el sesgo realizado.

3.2.6. Criterio para la objetividad

Algunos de los textos existentes en el corpus no son opinados ni expresan una subjetividad, como la opinión 3.12.

Diputado dice que comunas pueden prohibir la minería
el diputado (3.12)

Esto se debe a que, pese a que BuscOpiniones es un sistema que pretende extraer únicamente opiniones de los artículos de prensa, comete errores como todo sistema automático. El problema de determinar automáticamente si un texto es o no opinado no es sencillo y es en sí mismo un área de investigación en el Procesamiento de Lenguaje Natural.

Por esta razón se descartó realizar un filtrado previo de textos no opinados, ya que no solamente podría haber sido un proyecto en sí mismo, sino que además sería duplicar la funcionalidad de BuscOpiniones.

Ante la necesidad de clasificar de alguna forma este tipo de textos no opinados, se podría haber considerado el clasificarlos a todos como neutrales, ya que es en principio la categoría más similar a "objetivo" de las tres disponibles.

Sin embargo, existen algunos textos que pese a ser objetivos son claramente positivos, o claramente negativos como en este caso:

«Un brusco incremento». El diario informó este martes que las muertes se triplicaron en el Hospital Maciel en 2011 , de acuerdo al acta de inspección a la que tuvo acceso.
diario El Observador (3.13)

Aquí se informa un dato concreto y objetivo, por otro lado el dato del aumento por tres de la cantidad de muertes es claramente negativo en cualquier contexto.

Si considerásemos estas oraciones como neutras, no solamente confundirían al clasificador agregando ruido innecesario a los datos, sino que incluso seguramente el consumidor del producto no coincidiría con el resultado obtenido.

Por lo tanto, el criterio elegido para este tipo de textos es considerarlos neutrales siempre y cuando, pese a ser objetivos, no tengan una marcada positividad o negatividad; y en ese caso marcarlas como positivos o negativos respectivamente.

3.3. Dificultades

Es común utilizar el análisis de sentimiento en contextos donde se sabe previamente que el texto es subjetivo, como pueden ser blogs, calificaciones, recomendaciones de productos, o algunas redes sociales. Estos contextos ofrecen ventajas que ayudan a poder realizar un análisis. Es posible incluso conocer, además del carácter subjetivo de los textos, la fuente (usualmente determinada por quien lo publica), el tema mediante el conocimiento del lugar donde fue publicado (por ejemplo si es una revisión sobre un producto o servicio), y hasta la polaridad para generar un corpus si tienen estrellas u otro tipo similar de clasificación.

En el caso de este proyecto, se trabaja con opiniones de prensa que no cuentan con las ventajas mencionadas. Las opiniones son extraídas de los artículos, determinando la fuente y el alcance de las mismas en el texto. El tema de la opinión no es explícito, y debe realizarse un análisis para poder determinarlo. Cada uno de los puntos mencionados constituye un área dentro del análisis de sentimiento y la realización de los mismos conlleva un estudio específico de alta complejidad. Estos problemas son resueltos por BuscOpiniones, en etapas previas a este proyecto. Sin embargo, los errores cometidos por este programa pueden influir negativamente en la entrada del que se construye en este proyecto.

Incluso, normalmente las noticias y artículos de prensa son redactados en un lenguaje formal y cuidadoso en cuanto a la subjetividad. Se prioriza la exposición de la información ante la opinión y discusión de la misma. Esto no quita que exista texto opinado dentro de los documentos publicados, ya que es común encontrar citas de opiniones realizadas por personas como políticos, artistas, deportistas, etc; además de artículos de opinión, principalmente en los editoriales.

Incluso cuando existe texto subjetivo, es común encontrar opiniones solapadas o muy cautelosas, mucho más que en foros u otras procedencias de las opiniones, como se puede notar en la opinión 3.14:

> *«Estamos contentos y conformes por lo logrado en el Mundial de Sudáfrica y en la Copa América, pero se debe dar vuelta a la página y empezar muy concentrados en las eliminatorias», afirmó Tabárez.* (3.14)
>
> Tabárez

Tabárez está opinando muy positivamente, pero por prudencia baja la intensidad de la polaridad de la opinión. Este tipo de recursos son muy comunes en la prensa y son difíciles de captar automáticamente.

Otro aspecto importante es el idioma de texto a ser analizado. El inglés es el idioma más utilizado en la investigación de Procesamiento de Lenguaje Natural y consecuentemente es el que tiene una mayor cantidad de herramientas disponibles para realizar estudios sintácticos y semánticos sobre las opiniones. En este caso las opiniones son en idioma español, que tiene una cantidad mucho menor de herramientas, muchas de las cuales que no son tan efectivas, ya sea por la dificultad que tiene el lenguaje en sí o por el desarrollo de las mismas.

Capítulo 4

Enfoque de la solución

El objetivo planteado en este proyecto es construir un sistema que logre detectar la polaridad en opiniones. Concretamente, dada una opinión, devolver si esta es positiva, negativa o neutra. En este capítulo se resume la idea general de la implementación de la solución.

Como se expuso en la sección 2.3, la mayoría de los sistemas que detectan polaridad en textos se dividen en tres clases a partir de su construcción: métodos basados en reglas, métodos de aprendizaje automático e híbridos.

Los métodos de aprendizaje automático son los más utilizados últimamente en el área generando buenos resultados. Sin embargo, requieren para ello de contar con un corpus de entrenamiento lo suficientemente amplio como para abarcar todos los ejemplos de los cuales se aprende.

Por otro lado los sistemas basados en reglas no necesitan de un corpus de entrenamiento porque son las reglas y lexicones de palabras que influyen en la polaridad los que aportan el conocimiento, en lugar de aprender de los datos. Sin embargo, requieren de un corpus para analizar los resultados de los algoritmos. A su vez el lenguaje natural es complejo y no es posible modelarlo con facilidad.

Por estas razones, se decide explorar ambas alternativas ya que pese a que las dos tienen inconvenientes, ambas son viables y se ajustan al problema. También se implementa una solución híbrida que alimenta al aprendizaje automático con la información generada por las reglas en la búsqueda de aprovechar las mejores características de cada uno de los sistemas.

4.1. Sistemas

- **Sistema basado en reglas:**

El sistema basado en reglas determina la orientación semántica de una oración a partir de un lexicón sentimental que contiene las polaridades a priori de las palabras. Además, a partir de un análisis sintáctico de la oración, detecta y cuantifica fenómenos como la negación, intensificación, y la presencia de oraciones subordinadas adversativas, que influyen en la determinación de la polaridad general de la oración.

- **Sistema basado en aprendizaje automático:**

  Como se explicó anteriormente, para lograr crear un sistema que clasifique opiniones basado en aprendizaje automático fue necesario anotar un corpus de entrenamiento.

  El preprocesamiento de las opiniones genera tuplas basadas en las palabras que aparecen. Además, se agregan features basadas en el conteo de palabras positivas y negativas, agrupadas por categoría gramatical.

  Los algoritmos de aprendizaje utilizados son Multinomial Naive Bayes y Support Vector Machines, dos de los más utilizados en el área.

- **Sistema híbrido:**

  Recibe como features no solamente la información del clasificador explicado anteriormente, sino también la generada por los métodos de reglas.

  El objetivo es que las nuevas features le permitan aprender de las relaciones sintácticas entre los constituyentes de las opiniones que fueron analizadas por las reglas, así como del resultado del sistema de reglas.

  Como fue explicado en la sección 2.2, la negación es el principal elemento que cambia la polaridad de un término. Por lo tanto, la información sobre si una palabra está negada es una de las principales modificaciones que se agregan a la entrada del clasificador.

La implementación de la solución del problema se detalla en el capitulo 6.

4.2. Recursos necesarios

Para poder resolver los problemas planteados y generar las soluciones deseadas, es necesario contar con una serie de recursos:

- **Lexicón sentimental:** Un diccionario de palabras con sus polaridades para utilizar tanto en el sistema de reglas como en el de aprendizaje automático para obtener la orientación semántica de las palabras.
- **Tokenizador, segmentador y etiquetador morfosintáctico:** Utilizado para preprocesar las opiniones, obteniendo para cada una de ellas una lista de sus oraciones; para cada oración una lista de palabras; y para cada palabra su lema y rol morfosintáctico.

- **Parser sintáctico de dependencias:** Utilizado para obtener el árbol de dependencias para el análisis del sistema de reglas.
- **Corpus:** Es necesario un corpus anotado para realizar pruebas y obtener métricas de funcionamiento de los clasificadores. Además, se requiere un corpus anotado para el entrenamiento y las evaluaciones de resultados.

4.3. Corpus

Al no contar previamente con un corpus que se ajuste a las necesidades del problema, como primera instancia en este proyecto se decide crear un corpus anotado de opiniones.

Se consideró fundamental que el corpus se tratara de opiniones de prensa, prioritariamente temas políticos, deportivos, o de actualidad, como oposición a las reseñas que son una fuente habitual de corpus. Por estas razones, es difícil encontrar un corpus en español que a su vez cumpla con estas características, tanto que de hecho no fue posible encontrar uno.

Es por ello que fue necesario generar un corpus mediante la anotación de oraciones. Las oraciones de BuscOpiniones cumplen con el requisito fundamental de que son en idioma español, y del mismo contexto que las del problema a resolver. Por estas razones, se decidió generar el corpus mediante la extracción de oraciones de BuscOpiniones. La creación y análisis del corpus se describe en el capítulo 5.

Capítulo 5

Corpus

La correcta elección, recolección y utilización de un corpus, tanto de entrenamiento como de prueba, es una de las partes más importantes en cualquier problema del Procesamiento de Lenguaje Natural. En este caso, el conjunto de textos opinados sobre los que entrenar, en el caso del aprendizaje automático, y evaluar el algoritmo propuesto consistió en una de las tareas sustanciales del proyecto.

Debido principalmente a la escasez de recursos y corpus en español, se consideró de gran utilidad, no solo a nivel de este proyecto, generar y anotar un corpus de opiniones de prensa en español.

Este capítulo presenta una descripción del corpus generado, las decisiones llevadas a cabo, el análisis del mismo y la concordancia de sus anotadores.

Recolección de opiniones

Se seleccionaron en total 3000 opiniones a través de la realización de consultas en el sistema BuscOpiniones sobre las siguientes fuentes: Mujica, Astori, Tabaré Vázquez, Lacalle, Bordaberry, Tabárez, Suárez y Forlán. Estas son las fuentes que presentan más opiniones en la base de datos de BuscOpiniones. Fueron descartadas 801 opiniones del total ya que eran oraciones incompletas, como por ejemplo: «Forlán dijo», «Batlle dijo que el acuerdo de Uruguay», «Batlle afirmó que».

5.1. Anotación del corpus

Luego de obtenidas las oraciones se procedió a la anotación: el proceso de marcar cada una de ellas manualmente como positiva, neutral o negativa, además de descartar las que por algún motivo no debieran pertenecer al corpus.

Para facilitar este proceso, se creó una web privada para usuarios (anotadores), en la que se puede etiquetar de forma sencilla e intuitiva. La interfaz ofrece la posibilidad de etiquetar cada opinión del corpus o borrarla si está malformada o incompleta. Cada etiquetado se guarda en una base de datos junto con los del resto de los anotadores.

5.1.1. Anotadores

Antes de comenzar a anotar, fue necesario resolver la cantidad de anotadores necesarios para lograr un corpus completo y correctamente anotado.

Se consideró el escenario de hacer pública la página web y permitir que cualquier persona pudiera etiquetar las opiniones. De esta manera se podía distribuir el trabajo de la anotación y conseguir una mayor cantidad de opiniones etiquetadas.

Sin embargo, luego de la etapa de análisis de las opiniones del corpus extraído de BuscOpiniones, explicado con detalle en la sección A.2, se observó que muchas de las opiniones son complejas de etiquetar. Esta complejidad hace que para lograr que cualquier usuario etiquete las opiniones sea necesario que realice una etapa de capacitación previa, donde se unifiquen los criterios de la anotación. En caso contrario, cada anotador elegiría un camino distinto para la resolución de estos problemas, conllevando un alto riesgo de que exista mucha diferencia de criterios en las anotaciones.

Al ser una condición necesaria que los anotadores entiendan bien el problema, se decidió mantener la web privada, y que la anotación sea llevada a cabo únicamente por los integrantes del proyecto, aprovechando el conocimiento del problema generado en la etapa inicial del proyecto.

5.1.2. Método de anotación

El etiquetado del corpus de prueba se realiza con un etiquetado en base a la votación de los tres anotadores: a cada opinión se le asigna la etiqueta por mayoría. En caso de haber empate (debido a que los tres anotadores etiquetaron distintas categorías para la opinión), se la etiqueta como “empate”, para luego ser analizada con más detalle y se elige por consenso una etiqueta.

Esta forma de anotación tiene como ventaja que se genera una mayor certeza en la decisión de la categoría de la opinión, ya que se tiene en cuenta la opinión de todos los anotadores. Además, se puede realizar análisis de concordancia entre ellos. Por otro lado, etiquetar con este método resulta costoso en términos de tiempo.

Por esa razón, el corpus de entrenamiento es etiquetado por un único anotador. Con este método se acelera el proceso de anotación, algo primordial considerando que la rigurosidad no es tan imprescindible como en el corpus de prueba y sí es importante conseguir muchas opiniones anotadas.

Se decide entonces anotar en total 938 opiniones para el corpus de prueba y las restantes 1261 que sean anotadas por una única persona para utilizarlas como entrenamiento.

5.1.3. Criterios de anotación

Los textos extraídos de prensa son complejos de analizar debido a que la polaridad no es clara en muchas ocasiones, entre otras dificultades explicadas en los capítulos 2 y 3. Por lo tanto fue necesario definir criterios en la anotación previos a la anotación del corpus. A continuación se enumeran los distintos criterios llevados a cabo. Estos se encuentran explicados en detalle en la sección 3.2.

- **Criterio para la dependencia del contexto**: En el caso de la falta de contexto, se definió que las opiniones sean consideradas junto con su contexto social, cultural y político, ya que es justamente lo que se busca al realizar un proyecto con opiniones locales. Por otro lado, la expansión de contexto temporal en el texto se evaluó como fuera del alcance del proyecto, por lo que se considera únicamente la información que está presente en el texto recuperado y no la que lo acompañaba en el artículo original.
- **Criterio para múltiples polaridades**: El criterio definido es que se tome la sensación general que deja, fijándose en la más importante de las polaridades que expresa la fuente. La importancia puede medirse ya sea porque es sobre el tema más relevante, porque es la que destaca la fuente mediante los conectores que utiliza, porque es la más intensa de las presentes, o por cualquier otra razón.
- **Criterio para el tema incierto**: Para estos casos, se decide tomar como tema de la opinión para su análisis el que más se ajuste al real, permitiendo los que son subtemas del que fue buscado, e incluso en los casos en los que el tema es directamente distinto. Se considera toda la información disponible en la opinión.
- **Criterio para la ironía**: Este proyecto no abarca el problema de detectar ironía, por lo que seguramente este tipo de opiniones generen problemas para su correcta clasificación. Sin embargo, se decide anotar con la polaridad que realmente poseen, teniendo en cuenta el tono irónico.
- **Criterio para el sesgo de objetividades**: En estos casos es el sesgo de la selección lo que hace que la fuente exprese un estado privado subjetivo, y se decide considerarlos como opiniones con una polaridad marcada por el sesgo realizado.

- **Criterio para la objetividad**: Las opiniones objetivas se consideran neutrales siempre y cuando, pese a ser objetivos, no tengan una marcada positividad o negatividad. De otra forma marcan como positivos o negativos respectivamente.

Descripción del corpus

El corpus se compone de 2199 opiniones distribuidas como se muestra en el cuadro 5.1.

	Entrenamiento	Prueba	Total
Positivos	401	414	815
Neutrales	328	258	586
Negativos	532	266	798
Total	1261	938	2199

Cuadro 5.1: Cantidad de opiniones del corpus según el conjunto de entrenamiento y de prueba; agrupado por polaridad

5.2. Análisis de la anotación

En esta sección se detallan las conclusiones obtenidas al analizar los resultados de la etapa de anotación. Se describen las métricas de evaluación y los resultados tanto de la concordancia entre los anotadores, como de datos adicionales que se consideraron importantes para la conclusión de la calidad del corpus generado.

5.2.1. Concordancia entre anotadores

Como se explicó anteriormente, el corpus de prueba fue anotado manualmente con un método de votación entre los resultados de los tres anotadores. Es importante analizar qué concordancia hay entre los anotadores, es decir, qué tan de acuerdo están a la hora de anotar; por lo tanto se analizará en profundidad la concordancia entre los anotadores sobre el corpus de prueba.

Se propone utilizar para el análisis las medidas kappa de Fleiss y Cohen, así como accuracy, precision, recall y medida F1. A continuación se describe cada una de las medidas junto con la explicación del significado de los resultados que brindan.

5.2.1.1. Accuracy, Precision, Recall y F1-score

El resultado del análisis de concordancia entre anotadores, en cuanto a accuracy, precision, recall y F1 score, brinda una medida de tope en la eficacia del clasificador. Cuanto más lejano

del 100 % sean los resultados, significa que más complejas de clasificar son las oraciones, ya que existe más discordancia entre humanos, como se explica en la sección 7.2.2.

En esta sección se explican las métricas utilizadas y se muestran los resultados obtenidos para cada par de anotadores, así como el promedio final de las métricas.

La idea para el cálculo de éstos en cuanto a concordancia de anotadores, se trata a un anotador como *gold standard* y de esta forma se mide el grado en que el otro anotador se desvía de esta referencia.

Verdaderos y falsos positivos y negativos

Muchas de las medidas se definen a través de la clasificación de los resultados relativos a una de las clases del clasificador en cuatro categorías:

- **Verdaderos Positivos (VP):** Las muestras (opiniones en el caso de este proyecto) que son de la categoría que se está analizando y fueron correctamente clasificadas.
- **Verdaderos Negativos (VN):** Las muestras que no pertenecen a la categoría que se analiza y fueron correctamente clasificadas como no pertenecientes a ella.
- **Falsos Positivos (FP):** Las muestras que pertenecen a otra categoría pero fueron incorrectamente clasificadas como de la que se está analizando.
- **Falsos Negativos (FN):** Las muestras que no pertenecen a la categoría que fue analizada pero se clasificaron incorrectamente como pertenecientes a ella.

Es importante destacar que la notación de positivo o negativo en verdaderos/falsos positivos/negativos no se refiere a la clase en sí (que en nuestro caso son positivo, negativo y neutro) si no, así como muestran las definiciones, refiere a si la muestra fue clasificada dentro de la clase que debería haberlo sido.

Accuracy

La medida de accuracy (acierto) es la más básica de las existentes: es el porcentaje de muestras correctamente clasificadas.

Formalmente, se define como:

$$\frac{VP + VN}{VP + VN + FP + FN}$$

Es común pensar que el accuracy es suficiente como medida de evaluación, ya que permite conocer la cantidad de ejemplos bien y mal clasificados. Sin embargo, no siempre es una buena métrica: existen casos en los que puede dar muy buenos resultados en un clasificador cuya

utilidad es nula. Por ejemplo, en el caso de un corpus muy desbalanceado donde una clase A forma el 99 % y otra clase B el 1 % del mismo, podríamos hacer un clasificador que siempre predice A, sin importar cómo sea la muestra. Esto generaría un 99 % de accuracy, dando la impresión de tener un excelente resultado, aunque en realidad es absolutamente inútil.

Por esta razón, se toman en cuenta otras métricas que se describen a continuación.

Precision y Recall

Conocidas en español como precisión y exhaustividad, son medidas que permiten conocer la efectividad de un clasificador con respecto a una de las clases que lo conforman.

Precision es la métrica que presenta la relación entre la cantidad de muestras correctamente clasificadas como de una clase con todas las que fueron clasificadas como pertenecientes a la misma. Formalmente se define como:

$$\frac{VP}{VP + FP}$$

Cuanto más cercano al 1 sea el resultado, más confiable y certero es el clasificador cuando clasifica una muestra como de la clase analizada.

Normalmente, esta métrica se acompaña del *recall*, que presenta la relación entre la cantidad de muestras clasificadas como de una clase con la cantidad total de muestras de esa clase presente en el conjunto de test. Formalmente:

$$\frac{VP}{VP + FN}$$

Cuanto más cercano a 1 sea el recall, menor cantidad de ejemplos de una determinada clase serán "perdidos" al ser clasificados como de otra. En el ejemplo de la sección anterior de un corpus desbalanceado, el recall de la clase B (la que representa el 1 % del total) será seguramente la medida más importante a tener en cuenta.

Medida F

Precision y recall son medidas complementarias. Para generar una medida única que permita obtener la información de ambas, se crea la medida F que se define como la media armónica de las anteriores:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

Como puede verse, el parámetro beta se utiliza para ponderar la importancia de una de las dos medidas sobre la otra.

En el caso de este proyecto, que tiene clases balanceadas, se decide utilizar beta igual a uno, dando la misma importancia a ambas.

Resultados

Para facilitar la exposición de los resultados se nombra a los tres anotadores como A1, A2 y A3. Las tablas 5.2, 5.4 y 5.6 muestran las matrices de confusión entre los pares de anotadores: A1-A2, A2-A3 y A1-A3 respectivamente. A su vez, se presentan las métricas de precision, recall, accuracy y medida F1 para cada uno en las tablas 5.3, 5.5 y 5.7.

	Negativo	Neutral	Positivo
Negativo	**195**	38	19
Neutral	46	**134**	128
Positivo	12	25	**341**

Cuadro 5.2: Matriz de confusión entre A1 y A2

	Precision	Recall	F1-score
Negativo	**0.77**	0.77	0.77
Neutral	0.68	0.44	0.53
Positivo	0.70	**0.90**	**0.79**
Promedio	0.71	0.71	0.70
Accuracy total: **0.71**			

Cuadro 5.3: Resultados entre A1 y A2

	Negativo	Neutral	Positivo
Negativo	**200**	48	5
Neutral	46	**133**	18
Positivo	28	137	**323**

Cuadro 5.4: Matriz de confusión entre A2 y A3

	Precision	Recall	F1-score
Negativo	0.73	**0.79**	0.76
Neutral	0.42	0.68	0.52
Positivo	**0.93**	0.66	**0.77**
Promedio	077	0.70	0.72
Accuracy total: **0.70**			

Cuadro 5.5: Resultados entre A2 y A3

	Negativo	Neutral	Positivo
Negativo	**199**	45	8
Neutral	49	**207**	52
Positivo	26	66	**286**

CUADRO 5.6: Matriz de confusión entre A1 y A3

	Precision	Recall	F1-score
Negativo	0.73	**0.79**	**0.76**
Neutral	0.65	0.67	0.66
Positivo	**0.83**	0.76	0.74
Promedio	0.74	0.74	0.72
Accuracy total: **0.74**			

CUADRO 5.7: Resultados entre A1 y A3

Como se puede observar, los pares de anotadores A1-A3 obtuvieron mayor mejores resultados, sin embargo, comparando las tablas los tres pares de anotadores cuentan con patrones similares: existe poco acierto en lo que respecta la anotación de neutros, mostrando valores incluso por debajo del 50 %, pero un muy alto grado de concordancia en la anotación de positivos, arrojando valores hasta del 93 % de precisión por ejemplo entre los anotadores A2-A3; así como buenos valores en lo que respecta a los negativos.

La mayor diferencia entre A1-A3 y los demás, el cual es el causante de mejores resultados, son los valores de precision, recall y F1-score marcados sobre la anotación de opiniones neutras, ya que estas se encuentran por encima del 60 %.

En la tabla 5.8 se muestran los valores promedio de las métricas obtenidas anteriormente: accuracy, precision, recall y medida F1. Estos datos, como se explicará con más detalle en la sección 7.2.2, reflejan una cota superior de los resultados posibles que puede obtener el clasificador automático a desarrollar.

Métrica	Valor
Accuracy	0.71
Precision	0.74
Recall	0.72
F1-score	0.72

CUADRO 5.8: Promedios de las métricas de concordancia entre anotadores

5.2.1.2. Medida kappa de Cohen y Fleiss

La medida kappa de Fleiss (1971) [10] se utiliza para evaluar la concordancia entre N anotadores de un corpus, mediante el cálculo del grado de acuerdo en la clasificación, comparándose con el azar, siendo 1 el máximo valor posible. Por su parte la medida kappa de Cohen es una

medida similar pero únicamente para N=2 (dos anotadores). Al contarse con 3 anotadores, se tomará el promedio de las medidas entre pares de anotadores.

Se define κ de Cohen como

$$\kappa = \frac{p_0 - p_e}{1 - p_e},$$

donde p_o es el acuerdo observado relativo entre los evaluadores, o sea la suma de acuerdos sobre el total de anotaciones, y p_e es la probabilidad hipotética de acuerdo por azar, utilizando los datos observados para calcular las probabilidades de que cada observador clasifique aleatoriamente cada categoría. Si los evaluadores están completamente de acuerdo, entonces $\kappa = 1$. Si no hay acuerdo entre los evaluadores distinto al que cabría esperar por azar (según lo definido por p_o), $\kappa = 0$.

Definimos κ de Fleiss como

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e},$$

donde $\bar{P}$ se define como un promedio

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} P_i,$$

en donde N es la cantidad de ejemplos clasificados y P_i es la fracción de pares de anotadores que coinciden en una categoría en el ejemplo i-ésimo

$$P_i = \frac{1}{n(n-1)} \sum_{j=1}^{k} n_i j(n_i j - 1),$$

siendo n la cantidad de anotaciones que tuvo cada instancia, k la cantidad de categorías y $n_i j$ la cantidad de anotadores que asignaron el ejemplo i a la categoría j (observar que $\sum_{j=1}^{k} n_i j = n \forall i \in 1, ..., N \forall j \in 1, ..., k$).

$\bar{P}_e$ es similar a $\bar{P}$, pero aplicado a obtener pares de coincidencias en categorías al azar

$$\bar{P}_e = \sum_{j=1}^{k} p_j^2,$$

donde p_j es la proporción de las anotaciones que fueron a la categoría j

$$p_j = \frac{1}{Nn} \sum_{i=1}^{N} n_i j.$$

En el cuadro 5.9 los resultados del cálculo de los respectivos κ.

Anotadores	κ Cohen	κ Fleiss
A1-A2	0.56	*N/A*
A2-A3	0.54	*N/A*
A1-A3	0.60	*N/A*
A1-A2-A3	*N/A*	0.56

CUADRO 5.9: Medidas de Cohen κ y Fleiss κ de los anotadores

Existe poco acuerdo entre autores sobre qué valores kappa son considerados un buen resultado [11]. Sin embargo, se puede asegurar que valores cercanos o menores a cero implican un acuerdo pobre, mientras que los cercanos a 1 son considerados buenos. Se considera entonces que los valores obtenidos en este caso indican una concordancia de nivel moderado, ya que se consigue un promedio de 0.56 en ambos κ: promedios del de Cohen entre pares de anotadores, y Fleiss tomando en cuenta los 3 anotadores.

5.2.2. Datos Adicionales

Aunque las métricas de kappa detalladas anteriormente logren reflejar la calidad del corpus, se considera de gran valor el análisis exhaustivo de las diferencias entre las anotaciones.

Cabe destacar que existen distintos grados de discordancia en el sentido de que existe mayor diferencia entre etiquetas. Etiquetar neutral contra negativo (o positivo), se considera más discordante que etiquetar positivo contra negativo, aunque de hecho ambas generen una discordancia entre anotadores. En esta sección se analizará con detalle la discordancia entre los anotadores, para lograr comprender las diferencias.

En la tabla 5.10 se muestran los resultados de las cantidades y proporciones sobre las categorías de posibles resultados de las anotaciones y sobre el total de las muestras. Las posibles categorías son: los tres anotadores coincidieron, hubo 2 anotaciones iguales y una distinta o los tres discreparon.

Como se puede observar, para el caso en que los tres anotadores coinciden, se obtuvo un mayor porcentaje de negativos (52.6 %), luego de positivos (31 %) y por último neutral (16.4 %). Esto muestra por un lado el alto grado de acierto entre los anotadores al anotar opiniones negativas y a su vez que existe una gran dificultad para distinguir neutralidad en las opiniones.

Por otro lado, se observa un alto grado de discordancia en opiniones que fueron anotadas como positivas y neutras. Más de la mitad (52,2 %) de las opiniones con votación dividida

Categoría	Subcategoría	Cantidad	% en la categoría
Todos igual	Todos Positivos	125	31.0
	Todos Neutrales	66	16.4
	Todos Negativos	211	52.6
Votación 2 a 1	2 a 1 Neu-Pos	228	52.2
	2 a 1 Neu-Neg	81	18.5
	2 a 1 Pos-Neg	128	29.3
Todos distinto	N/A	39	N/A

CUADRO 5.10: Resultados de las anotaciones por subcategorías

fueron anotadas con esta distribución. Sin embargo, existe poca discordancia en opiniones entre negativos y neutrales, obteniendo solo un 18,5 % del total sobre esta categoría.

Por último, se obtuvo un muy bajo grado de discordancia total; solamente 39 opiniones, por lo que en conclusión, a pesar de existir una alta complejidad en la determinación de polaridad en opiniones de textos de prensa, se logra mantener una cierta concordancia en la anotación.

Capítulo 6

Solución del problema

En este capítulo se describe en detalle la solución llevada a cabo para el problema dado. Se muestran las herramientas utilizadas, las líneas bases elegidas y se describen los tres sistemas generados: uno basado en reglas, uno en aprendizaje automático y uno híbrido.

6.1. Herramientas utilizadas

Dada la complejidad que conlleva la tarea de analizar lenguaje, es natural que para investigar nuevos elementos del mismo se haga uso de herramientas desarrolladas previamente tales como analizadores morfosintácticos, corpus anotados, parsers, etc. Los métodos supervisados requieren de conjuntos, del orden de miles hasta centenas de miles de datos etiquetados previamente, cuya construcción requiere en general de proyectos exclusivos. Por otro lado los métodos basados en reglas, en la gran mayoría de los casos, requieren lexicones sentimentales que terminan siendo claves en los resultados obtenidos y cuya construcción también es una tarea de gran envergadura. Es por estas razones que la reutilización de herramientas de análisis de lenguaje desarrolladas previamente es crucial para poder realizar un proyecto de investigación.

Lexicón sentimental

El diccionario de palabras SODictionariesV1.11.Spa utilizado en [29] es un lexicón con valores que representan la polaridad de la palabra fuera de un contexto específico en una escala que va de -5 a +5. Las palabras neutrales, representadas con orientación semántica 0 fueron descartadas. El diccionario consta de cinco listas de palabras separadas en las categorías intensificadores y negadores, adjetivos, sustantivos, verbos y adverbios. Cada una de ellas se encuentra integrada por lemas de palabras junto con su valor sentimental. La lista de intensificadores, a diferencia de las otras, no contiene el valor de polaridad a priori de la palabra, sino el valor con el cual intensifica o niega.

Cada una de las palabras fue clasificada a mano por un anotador, con la consigna de que el valor atribuido a la palabra refleje tanto la polaridad a priori de la palabra, así como su intensidad, promediada entre las interpretaciones más probables de la misma. Originalmente el diccionario fue construido en inglés, luego traducido al español mediante el traductor de *Google*[1] y finalmente fue corregido manualmente. Los diccionarios fueron luego revisados por un comité de tres investigadores para minimizar la subjetividad que surge de clasificar la orientación semántica a mano.

El diccionario cuenta con un total de 4895 palabras: 2057 adjetivos, 594 adverbios, 1350 sustantivos, 758 verbos y 136 intensificadores y negadores.

Tokenizador, segmentador y etiquetador morfosintáctico

FreeLing[2] es una librería que ofrece herramientas para tareas relacionadas con el procesamiento del lenguaje natural. Está diseñada para ser utilizada como una librería externa desde cualquier aplicación que requiera de servicios como: tokenización de textos, etiquetado morfosintáctico, separación en oraciones, entre otros. En el contexto de este proyecto es utilizada como un módulo integrado a Python, y se hace uso específicamente de los módulos de tokenización, segmentación, y de etiquetado morfosintáctico. Con ellos se realiza el pre-procesamiento de las opiniones, obteniendo para cada token su lema y su rol morfosintáctico representado en una etiqueta EAGLE [3], que luego sirven de entrada para el parser de dependencias.

Parser sintáctico de dependencias

Mate Tools [6] es un parser sintáctico y semántico que, entre su pipeline de funciones, realiza parsing de dependencias. La versión específica utilizada en este proyecto fue desarrollada en [20], que utiliza el mismo parser desarrollado por (Bohent, 2009) entrenado con las particiones de entrenamiento de Tibidabo Treebank y IULA Spanish LSP Treebank. El parser recibe como entrada un archivo en formato CoNLL2009 generado a través de Freeling y devuelve un archivo en el mismo formato que además tiene la información del parsing efectuado.

Lenguaje de programación

Python fue lenguaje elegido para llevar a cabo este proyecto. La principal razón es que cuenta con una extensa sección de librerías de gran utilidad para la realización del proyecto como NLTK [4] para el procesamiento de lenguaje natural y Scikit-learn [5] para el aprendizaje automático. Además, se valora su legibilidad, mantenibilidad y simpleza.

[1] https://translate.google.com/
[2] http://nlp.lsi.upc.edu/freeling/node/1
[3] http://www.cs.upc.edu/ nlp/tools/parole-sp.html
[4] http://www.nltk.org/
[5] http://scikit-learn.org/stable/

6.2. Línea Base

Para conocer el punto de partida y determinar un mínimo con el que poder comparar los resultados obtenidos se definen 3 líneas base.

Aleatorio

Clasificador que selecciona aleatoriamente la clase que pertenece cada opinión, sin tomar en cuenta las palabras ni el contexto.

Todos negativos

Todos los ejemplos son clasificados por la clase más probable a priori. En el corpus de prueba de este proyecto, como se detalla en la tabla 5.1 la mayor cantidad de opiniones son negativas, por lo que se toma en cuenta un clasificador que clasifica todas las opiniones como negativas.

Sumatoria de polaridades

Es una técnica que modela un texto como una conjunto de palabras, sin tomar en cuenta el orden o las relaciones sintácticas. Dada una opinión el clasificador calcula la diferencia entre la sumatoria de los valores de polaridad de las palabras positivas y negativas que tiene. Para llevar esto a cabo, se utiliza el lexicón sentimental descrito en la sección 6.1 previo a las modificaciones realizadas en este proyecto.

Para este clasificador, se considera que una opinión es negativa, si la diferencia entre la sumatoria del grado de polaridad de todas las palabras positivas y negativas que se encuentran en la oración es menor que 0; como neutra si es igual a 0; y positiva si es mayor a 0. Se modificaron estos límites probando ampliar el rango de los neutros a [-1,1] y [-2,2] pero en esos casos se obtienen peores resultados.

6.2.1. Resultados

Los resultados de la clasificación de los clasificadores de línea base se reportan en las tablas 6.1, 6.2 y 6.3, para los clasificadores aleatorio, todos negativos, y sumatoria de polaridades respectivamente. Como se puede observar, la línea base que arroja mejores resultados es la sumatoria de polaridades.

	Precision	Recall	F1-score
Negativo	**0.42**	0.29	0.34
Neutral	0.25	**0.37**	0.30
Positivo	0.35	**0.37**	**0.36**
Promedio	0.36	0.33	0.34
Accuracy total: **0.33**			

Cuadro 6.1: Resultados de la línea base aleatoria

	Precision	Recall	F1-score
Negativo	**0.42**	**1.00**	**0.59**
Neutral	0.00	0.00	0.00
Positivo	0.00	0.00	0.00
Promedio	0.18	0.42	0.25
Accuracy total: **0.42**			

Cuadro 6.2: Resultados de la línea base todos negativos

	Precision	Recall	F1-score
Negativo	0.50	0.48	0.49
Neutral	0.44	0.29	0.35
Positivo	**0.52**	**0.66**	**0.58**
Promedio	0.48	0.49	0.48
Accuracy total: **0.50**			

Cuadro 6.3: Resultados de la línea base de suma de polaridades

6.3. Clasificador basado en reglas

En esta sección se describe la implementación del clasificador basado en reglas. Primero se presenta la arquitectura de la solución propuesta, detallando las funciones realizadas por cada uno de los módulos que la componen. Luego, se explica en detalle el algoritmo de clasificación desarrollado.

Los resultados obtenidos para el clasificador basado en reglas se muestran en detalle en la sección 7.2.3.

6.3.1. Arquitectura de la solución

El clasificador basado en reglas recibe como entrada una oración y devuelve como salida una etiqueta de orientación semántica: positiva, negativa o neutral.

Para lograrlo, se modela el clasificador como un sistema constituido por un conjunto de módulos con funciones independientes. Los módulos se encuentran en *pipeline*, lo que significa que la salida de cada módulo es la entrada del siguiente. La entrada del módulo inicial es la opinión mientras que la salida del módulo final es la orientación semántica resultante. En la figura 6.1

se ilustra el pipeline de funciones que se realiza en la solución propuesta. A continuación se detalla la función de cada módulo.

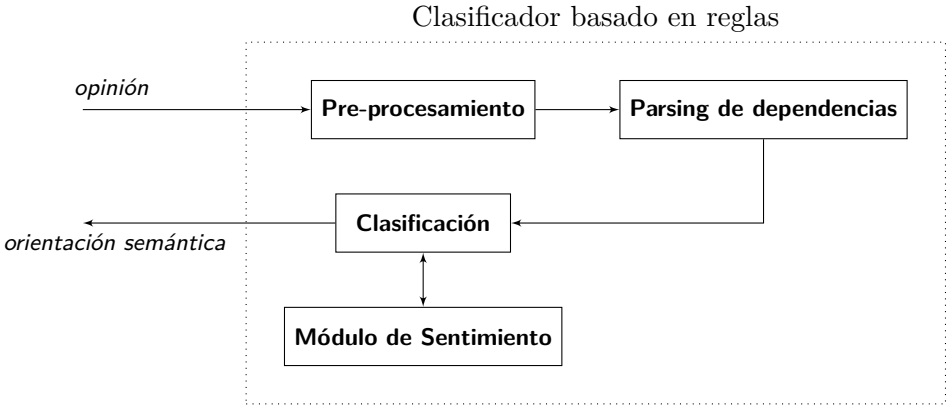

FIGURA 6.1: Arquitectura de la solución del clasificador basado en reglas.

6.3.1.1. Pre-procesamiento

Las opiniones extraídas de *BuscOpiniones* varían en cuanto a la cantidad de oraciones que las componen. Las herramientas de análisis que son utilizadas en módulos siguientes requieren de unificar el formato de las entradas antes de ser utilizadas. Para llevar la opinión al formato de entrada requerido en el módulo siguiente se deben realizar los procesos de *tokenización*, *segmentación* y *etiquetado morfológico*, sobre la opinión.

La *tokenización* es el proceso de separar el texto en *tokens*, donde un *token* es una cadena de caracteres con un significado. Cada símbolo ortográfico y cada palabra se consideran como un *token*. La *segmentación* se encarga de identificar y separar las distintas oraciones del texto tokenizado. La figura 6.2 muestra una aplicación de los procesos de *tokenización* y *segmentación* sobre el texto «*Juan corre. Sale al mediodía.*»

$$\{[(Juan), (corre), (.)], [(Sale), (a), (el), (medio), (día), (.)]\}$$

FIGURA 6.2: Ejemplo de *segmentación* y *tokenización* sobre el texto «*Juan corre. Sale al mediodia.*»

Para realizar el *etiquetado morfológico* se parte de la oración tokenizada y segmentada, y a cada token se le asigna una *etiqueta morfológica*.

Se define *etiqueta* como la información léxica que identifica a un token. Típicamente la información incluye la categoría gramatical, y otras características del token como el género, el número, el modo y el tiempo, dependiendo de la categoría.

En resumen, el módulo de pre-procesamiento recibe como entrada una opinión de *BuscOpiniones* y devuelve como salida una lista de oraciones *tokenizadas* con sus respectivos *etiquetados morfológicos*.

6.3.1.2. Análisis sintáctico de dependencias

El análisis sintáctico de dependencias es un proceso que permite obtener información sobre la estructura sintáctica de una oración, a partir de la información morfológica de sus palabras.

La información sintáctica se representa mediante conexiones binarias entre palabras llamadas *relaciones de dependencia*. A la palabra sintácticamente subordinada en una relación de dependencia se le denomina *dependiente* y a la palabra de la que depende, *padre*. El *tipo de dependencia* es la etiqueta que se le asocia a cada relación y que resume la información sintáctica que liga a la palabra subordinada con la subordinante. A la estructura resultante del análisis de dependencias se la denomina *grafo de dependencias*.

Sea $S = etiquetado(o_i) = [(w_1, e_1), .., (w_n, e_n)]$ una oración *tokenizada* y *etiquetada morfológicamente*, y sea $R = \{r_1, .., r_m\}$ un conjunto que representa los posibles tipos de relaciones de dependencia que pueden establecerse entre dos palabras de una oración, un *grafo de dependencias* $G = (V, A)$ es un grafo donde:

- $V = \{w_0, .., w_n\}$ es el *conjunto de nodos del grafo*, constituido por cada uno de los *tokens* pertenecientes a la oración, más w_0 que representa al nodo artificial $ROOT$.
- $A \subseteq (V \times R \times V)$ representa el *conjunto de relaciones de dependencias* para el análisis de una oración en particular.
- Cada uno de los elementos de A es una *relación*, $w_i \rightarrow w_j$, donde w_i es el padre y w_j es el dependiente, a la que se le atribuye un *tipo de dependencia* $r \in R$.
- G es un grafo *conexo*, *dirigido* y *acíclico*.
- Cada *nodo* perteneciente a G tiene un solo *padre*, y existe un nodo raíz sin padre ($ROOT$).

Las restricciones mencionadas en los últimos dos puntos implican que el grafo de dependencias es un *árbol*. Esto permite utilizar indistintamente los términos *grafo de dependencias* y *árbol de dependencias*, y en el marco de este proyecto se utiliza de aquí en más el segundo término. Además, al ser un *árbol* la estructura generada, para hacer referencia a la estructura se utiliza la terminología asociada a los árboles.

En la figura 6.3 se muestra un ejemplo de un árbol de dependencias para la oración «*Luis es un muy buen jugador de fútbol.*» En el ejemplo, el árbol de dependencias tiene como raíz al nodo *«ROOT»*, y como hijo al *árbol* de raíz *«es»*. Luego el *árbol* de raíz *«es»* tiene como hijos a los *nodos hoja «Luis»* y *«.»*, y al *árbol* de raíz *«jugador»*, y así sucesivamente.

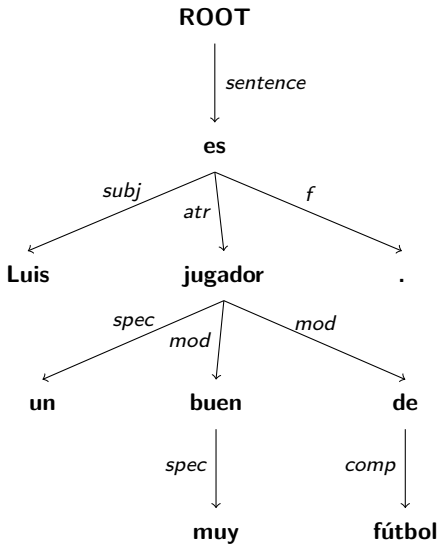

FIGURA 6.3: Ejemplo de un análisis sintáctico de dependencias.

6.3.1.3. Módulo de clasificación

El módulo de clasificación es el encargado de calcular la polaridad de la opinión. La entrada que recibe proviene del módulo de análisis de dependencias y es un conjunto de árboles de dependencias que tienen la información sintáctica y las etiquetas morfológicas para cada oración de la opinión. Dado que se cuenta con la información de *componentes* de la opinión (sus oraciones), y no con información que abarque *la totalidad* de la opinión, es importante determinar cómo se relaciona la polaridad total de la opinión con la polaridad de sus componentes.

El principio de composicionalidad establece que el significado de las expresiones complejas está determinado por el significado de las expresiones que la componen y la estructura que las relaciona. Siguiendo este razonamiento, se podría decir que la *polaridad* de expresiones complejas está determinada por la *polaridad de los componentes* que la componen y la *estructura que las relaciona.*

En este trabajo las expresiones complejas son *opiniones*, y sus componentes son *oraciones*. A partir del concepto anterior, se determinó que la polaridad total de una opinión sea la *suma* de las polaridades de cada una de las oraciones que la componen.

Existen casos en que hay oraciones que tienen más relevancia que otras, por ejemplo en las reseñas de películas usualmente se realiza una conclusión final que podría llegar a tener más peso del punto de vista de la polaridad total de la reseña. En el caso de las opiniones extraídas de *BuscOpiniones* no es claro que exista un fenómeno equivalente, y se concluyó que es razonable que todas la oraciones tengan el mismo peso.

El módulo de clasificación entonces, calcula la polaridad resultante de la opinión como la suma de las polaridades de las oraciones que la componen. En el algoritmo 6.1 se resume el pipeline de acciones que realiza el clasificador de reglas para calcular la polaridad resultante de la opinión. La función *clasificar_oración* y el algoritmo que utiliza para determinar la polaridad de cada oración se detalla en la sección 6.3.2.

Algoritmo 6.1: Algoritmo general del clasificador de reglas

Datos: opinión de BuscOpiniones O

Resultado: etiqueta de polaridad de la opinión e, donde $e \in \{Neutral, Positiva, Negativa\}$

$oraciones\_etiquetadas = preprocesar(O)$

$árboles = análisis\_de\_dependencias(oraciones\_etiquetadas)$

$sent = 0$

para $árbol \in árboles$ **hacer**
  $sent = sent + clasificar\_oración(árbol)$
fin

si $sent = 0$ **entonces**
  devolver $Neutral$
en otro caso
  si $sent > 0$ **entonces**
    devolver $Positiva$
  en otro caso
    devolver $Negativa$
  fin
fin

6.3.1.4. Módulo de sentimiento

El módulo de sentimiento no pertenece al *pipeline* de funciones de la solución, sino que se plantea como un módulo independiente que provee una interfaz con métodos para extraer información del lexicón de sentimiento. El lexicón utilizado en este clasificador consta de un conjunto de 6 listas de palabras: 4 de polaridades sentimentales (sustantivos, verbos, adjetivos y adverbios), una lista de negadores, y una lista de intensificadores. El lexicón utilizado es explicado con más detalle en la sección 6.1.

El objetivo principal del módulo es ofrecer funciones que, a partir de una palabra con su información morfológica, permitan determinar si representa un *negador*, *intensificador*, o si representa una palabra con una *polaridad sentimental* no neutra.

Además, ofrece funciones que permiten obtener valores que cuantifican la intensidad de los fenómenos de negación e intensificación, y permiten obtener la polaridad sentimental de la palabra.

En las *listas de polaridades sentimentales* cada palabra tiene asignada un valor de sentimiento s entero que cumple que,

$$-5 \leq s \leq 5, s \neq 0, s \in Z.$$

El valor absoluto del sentimiento representa la intensidad del mismo (5 es la máxima intensidad, y 1 la menor), mientras que el signo representa la polaridad (si el signo es negativo la palabra tiene sentimiento negativo y si el signo es positivo la palabra tiene sentimiento positivo).

La *lista de negadores* le atribuye a cada negador un nivel de intensidad n que cumple que,

$$1 \leq n \leq 5, n \in N.$$

La negación se representa con valores en esa escala ya que en este proyecto se decidió modelar la negación como un corrimiento hacia la polaridad inversa del resultado parcial, como fue propuesto en [29] y explicado en la sección 2.4. Sea p la polaridad parcial sobre la que tiene alcance el negador con intensidad n, la polaridad resultante con la influencia de la negación se calcula como,

$$p_{res} = p - sig(p) \cdot n,$$

donde $sig(p)$ es la función signo que devuelve 1 si $p > 0$, 0 si $p = 0$ y -1 en otro caso. Por ejemplo, para la oración «*Su conducta no es la mejor.*», si aplicamos la influencia del negador *«no»* con intensidad 5 a la polaridad sentimental +2 de *«mejor»*, el resultado es

$$2 - sig(2) \cdot 5 = -3.$$

La *lista de intensificadores* le atribuye a cada intensificador un nivel de intensidad i real que cumple,

$$-2 \leq i \leq 2, i \neq 0, i \in R.$$

Cuando un intensificador i actúa sobre un elemento con polaridad sentimental no neutral p, lo modifica de manera que la polaridad sentimental resultante p_{res} se calcula como,

$$p_{res} = p \cdot (1 + i),$$

donde, si $i > 0$ la palabra intensifica la polaridad y si $i < 0$ la disminuye. Por ejemplo, para la oración «*Su conducta es muy buena.*», si aplicamos la influencia del intensificador *«muy»* con intensidad 0,5 a la polaridad sentimental +2 de *«buena»*, el resultado es

$$2 \cdot (1 + 0{,}5) = 3.$$

A partir de las funciones ofrecidas por el módulo de sentimiento se presenta un algoritmo que dado un *nodo* del árbol de dependencias devuelve su valor de sentimiento, negación e intensificación. La función se detalla en el algoritmo 6.2.

Algoritmo 6.2: Algoritmo para analizar *nodo*

```
Función analizar_nodo(nodo)
    sentimiento = 0
    negación = 0
    intensificación = 0
    si es_negador(nodo) entonces
        negación = obtener_negación(nodo)
    en otro caso
        si es_intensificador(nodo) entonces
            intensificación = obtener_intensificación(nodo)
        en otro caso
            si tiene_sentimiento(raíz) entonces
                sentimiento = obtener_sentimiento(nodo)
            fin
        fin
    fin
    devolver sentimiento, negación, intensificación
end
```

6.3.2. Algoritmo de cálculo de polaridad de una oración

En esta sección se presentan una serie de algoritmos que plantean una solución al cálculo de la polaridad total de una oración a partir de las palabras que la componen y de su estructura sintáctica.

Primero se presenta un algoritmo de cálculo básico, que describe la manera en que se recorre de forma recursiva el árbol de dependencias y cómo se utiliza el módulo de sentimiento para obtener la polaridad a priori de las palabras que componen la oración. Luego, se muestran dos variaciones del algoritmo básico: una que contempla el fenómeno de la *negación*, y otra que contempla el de la *intensificación* y las *oraciones adversativas* de forma conjunta. Por último, se presenta un algoritmo completo que toma en cuenta todos los fenómenos léxicos mencionados.

6.3.2.1. Algoritmo básico

Cada árbol de dependencias consta de un *nodo* raíz y de sus *hijos* que pueden ser tanto *árboles*, como *nodos hoja*.

La polaridad de un *nodo hoja* se determina a partir de la palabra que representa y de su polaridad a priori extraída del lexicón sentimental. Por ejemplo, para un nodo hoja n que tiene como palabra a *«excelente»* la función $analizar\_nodo(nodo)$, devuelve como sentimiento $+4$.

Por otro lado, determinar la polaridad de un *árbol* no resulta tan simple ya que implica establecer un conjunto de reglas que refleje las relaciones sintácticas y semánticas entre la raíz y los hijos del árbol.

Un primer algoritmo básico consiste en tomar la polaridad total de un *árbol* como la suma de las polaridades de sus *hijos* y su *raíz*. Para cada hijo de la *raíz del árbol* se determina si es un *árbol* o es una *hoja*. En caso de ser *hoja* se obtiene del módulo de sentimiento su polaridad a priori, y se suma a la polaridad total del árbol analizado. En caso de que sea un *árbol* se ejecuta una llamada recursiva de la función sobre el mismo, obteniendo su polaridad. Luego, la polaridad del hijo *árbol* se suma a la polaridad total del *árbol padre*. Por último, se analiza la *raíz* del árbol, se suma su valor de polaridad a la polaridad total y se retorna el resultado. En el algoritmo 6.3 se presenta en detalle la función planteada.

Algoritmo 6.3: Algoritmo básico para calcular la polaridad total de un árbol

```
Función clasificar_árbol(árbol)
    r = 0
    para hijo ∈ obtener_hijos(árbol) hacer
        si es_árbol(hijo) entonces
            r = r + clasificar_árbol(hijo)
        en otro caso
            sent, neg, intensi = analizar_nodo(hijo)
            r = r + sent
        fin
    fin
    sent, neg, intensi = analizar_nodo(obtener_raíz(árbol))
    r = r + sent
    devolver r
end
```

En la figura 6.4 se muestra una aplicación del algoritmo básico sobre el árbol de dependencias de la oración *«Luis es un buen jugador de fútbol.»* La figura muestra la aplicación del algoritmo con una ilustración gráfica que se explica a continuación.

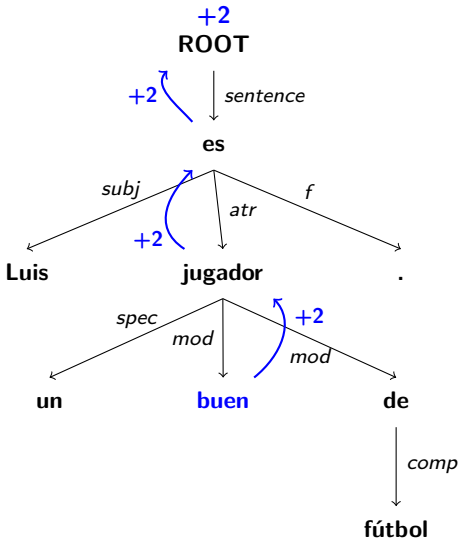

FIGURA 6.4: Ejemplo ilustrativo de la ejecución del algoritmo básico de clasificación de oración.

En el ejemplo, el primer árbol que se recorre es el de raíz *«ROOT»*, que al ser un nodo artificial no tiene polaridad. Luego, se analiza al único hijo del nodo raíz *«ROOT»* que es el árbol de raíz *«es»*. El primer hijo analizado del nodo *«es»* es el nodo hoja *«Luis»*, cuya polaridad es neutra. Luego se ejecuta una llamada recursiva sobre el árbol de raíz *«jugador»*. El único hijo del árbol de raíz *«jugador»* que tiene una polaridad no neutra es el nodo *«buen»*, que en el lexicón sentimental tiene una polaridad positiva, señalizada por el color azul (rojo en caso de negativa), de +2 y se suma al total del árbol. En la figura se representa la influencia de la palabra positiva *«buen»* como una flecha azul (ya que la polaridad es positiva, el rojo representa la polaridad negativa) que va desde el nodo *«buen»* hasta la raíz del árbol al que pertenece que es *«jugador»*, con una etiqueta de la polaridad que aporta, +2. Como el nodo *«un»* y el árbol de raíz *«de»* tienen polaridad neutra, el resultado que devuelve el árbol de raíz *«jugador»* es también $0 + 2 + 0 = +2$ y se representa con una flecha azul, que va desde *«jugador»* hasta su nodo padre *«es»*. Por último, como el nodo raíz *«es»* es neutro, el resultado final del análisis de la oración es +2 y se lo representa en el número encima del nodo artificial *«ROOT»*.

6.3.2.2. Algoritmo con negación

La negación es un fenómeno lingüístico que produce cambios en la polaridad contextual de las palabras sobre las que tiene alcance. La forma en que la negación cambia la orientación semántica y el alcance que el fenómeno tiene sobre el texto, son temas de gran interés en el área del PLN ya que tienen una alta complejidad. Esto deriva en que no sea fácil determinar un conjunto de reglas que modelen al fenómeno.

Para los algoritmos de análisis de sentimiento dos formas han sido las más utilizadas para modelar el efecto de la negación. Una de ellas es modelar los negadores como inversores de la polaridad. Con este modelo el efecto de la negación es invertir la polaridad sentimental de los elementos sobre los que tiene alcance. Como fue explicado en la sección 2.4 este modelo no siempre lleva a buenos resultados.

La otra forma consta de modelar el efecto de la negación como un corrimiento hacia la polaridad inversa de la polaridad de los elementos sobre los que tiene alcance, como se especifica en la sección 6.3.1.4. En [29] se prueban ambos métodos, y el modelado mediante corrimiento obtiene mejores resultados, por lo que es el que se decide adoptar en este proyecto.

Para determinar el alcance de la negación, en el marco de este proyecto, se hizo uso de la información sintáctica que brinda el árbol de dependencias. En el árbol los negadores usualmente aparecen con la etiqueta *mod* (modificador), y son subordinados por el elemento sobre el que tienen alcance.

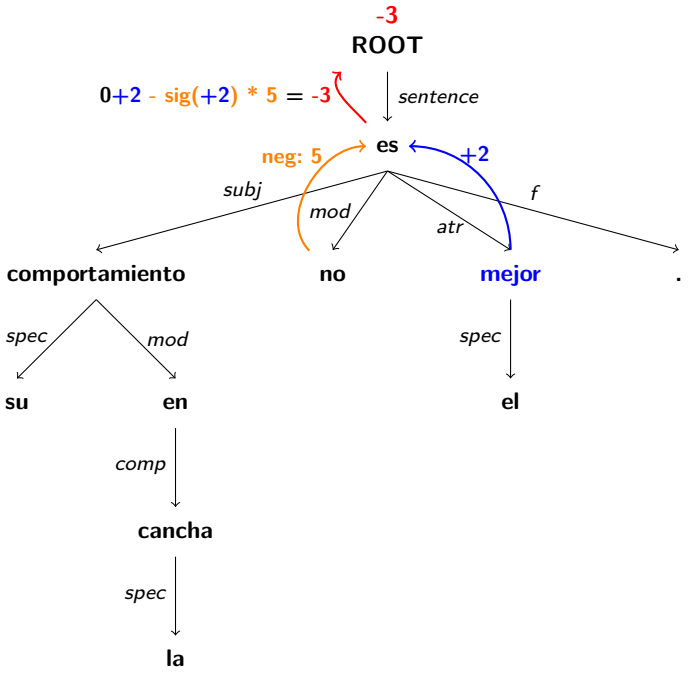

FIGURA 6.5: Ejecución ilustrativa del algoritmo de clasificación de oración con negación.

Un ejemplo de esto se puede ver en la figura 6.5, donde el nodo negador *«no»* está subordinado por el nodo padre *«es»*, en la oración *«Su comportamiento en la cancha no es el mejor.»* En este caso, *«es»* tiene polaridad neutra, por lo que una negación aplicada directamente sobre su polaridad no tendría efecto. Por otro lado, leyendo la oración se puede ver que la negación tiene alcance sobre la polaridad sentimental introducida por el sintagma adjetival *«el mejor»*. Esto se podría modelar aplicando la negación del nodo negador *«no»* sobre la polaridad total del árbol al que pertenece, que en este caso es el de raíz *«es»*. En el ejemplo se puede apreciar como actúa la negación sobre el resultado de la polaridad total del árbol +2 con el algoritmo

planteado en el módulo de sentimiento. Al resultado se le realiza un corrimiento de magnitud 5 hacia el polo sentimental contrario del resultado parcial $2 - sig(2)*5 = -3$, transformando el resultado positivo $+2$ en -3, el cual es consistente con la orientación semántica negativa de la oración.

Por otro lado, si aplicamos el mismo modelo sobre la oración *«Su comportamiento no es bueno y comete errores.»* el resultado no es el deseado. En la figura 6.6 se puede ver que en el árbol de dependencias, la oración subordinada por la conjunción *«y»* es un *árbol hijo* del árbol de raíz *«es»*. Siguiendo el modelo propuesto, el resultado total del árbol de raíz *«es»* es la polaridad positiva que introduce el *nodo hoja «bueno»* $+2$, sumada al resultado del árbol de raíz *«y»* -2. Si la negación se aplica sobre la suma $(2-2)*sig(2-2)*5 = 0$, se retora un resultado neutral. Este resultado no es consistente con la orientación semántica de la oración, y el problema reside en que el alcance planteado para la negación contempla también la oración subordinada por el conector coordinador *«y»*. Para solucionar el problema se plantea no aplicar la negación sobre los *árboles hijos* cuyo nodo raíz sea una conjunción, como se muestra en el ejemplo. Con un círculo color oliva se marca al nodo *«y»* que es una conjunción, y se representa su influencia sobre su nodo padre *«es»* con una flecha color oliva que tiene como etiqueta *conj:* y el resultado de la polaridad del árbol. Con el nuevo alcance planteado, la negación se aplica sobre el resultado parcial, sin contar el árbol de raíz conjunción *«y»*. Luego se suma el resultado del árbol de raíz *«y»*, retornando $2-sig(2)*5+(-2) = -5$, un valor consistente con la orientación semántica de la oración.

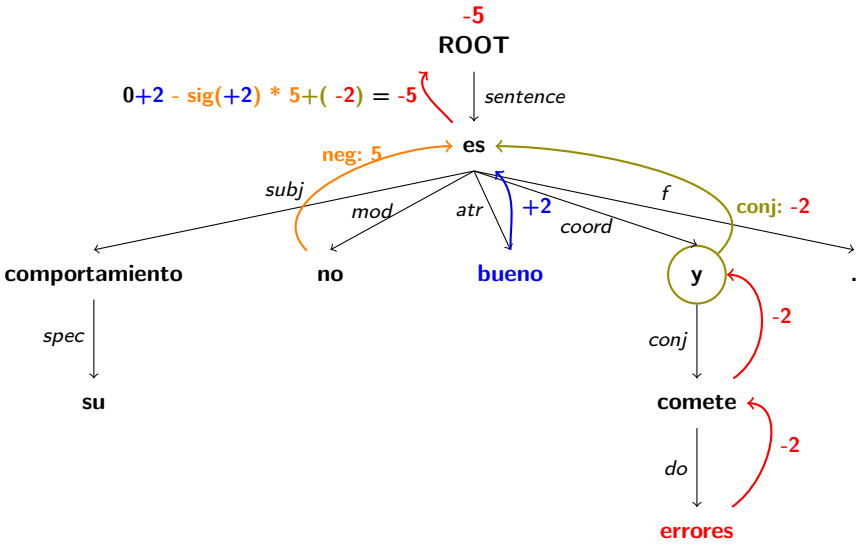

FIGURA 6.6: Ejecución ilustrativa del algoritmo de clasificación de oración con negación sin alcance sobre conjunciones.

En el algoritmo 6.4 se detalla el algoritmo final de negación que se aplica sobre una árbol de dependencias. Este, a diferencia del algoritmo básico, analiza para cada nodo hoja de cada árbol si es un negador. Si lo es, se suma el valor devuelto por el módulo de sentimiento a una

variable *negación*. En caso de encontrarse dos negadores sobre el mismo árbol, por simplicidad se utiliza el valor del último negador, y se invierte el sentido del corrimiento. Además, se llevan dos resultados r y r_conj para separar las polaridades de los árboles de raíz conjunción. A partir de la función $raiz\_conj$ se determina si la raíz de un *árbol hijo* es una conjunción. En caso de serlo, el resultado se suma a r_conj, de lo contrario se suma a r. Una vez analizados los hijos del árbol y su raíz, se aplica la negación con un corrimiento de la polaridad sobre r y se le suma r_conj. Por último se devuelve el nuevo resultado.

Algoritmo 6.4: Algoritmo para cálculo de polaridad con negación

Función *clasificar_árbol(árbol)*
- $negación = 0$
- $r = 0$
- $r\_conj = 0$
- **para** $hijo \in obtener\_hijos(árbol)$ **hacer**
  - **si** $es\_árbol(hijo)$ **entonces**
    - **si** $raiz\_conj(hijo)$ **entonces**
      - $r\_conj = clasificar\_árbol(hijo) + r\_conj$
    - **en otro caso**
      - $r = clasificar\_árbol(hijo) + r$
    - **fin**
  - **en otro caso**
    - $sent, neg, intensi = analizar\_nodo(hijo)$
    - $r = r + s$
    - **si** $negación \neq 0$ **entonces**
      - $negación = -negación$
    - **en otro caso**
      - $negación = neg$
    - **fin**
  - **fin**
- **fin**
- $senti, neg, intensi = analizar\_nodo(obtener\_raíz(árbol))$
- $negación = negación + neg$
- $r = r + senti$
- $r = r + signo(r) \cdot negación + r\_conj$
- **devolver** r

end

6.3.2.3. Algoritmo con intensificación

El fenómeno de la intensificación presenta desafíos similares al de la negación. Se debe determinar sobre qué elementos del texto tiene alcance y cómo afectan la polaridad sentimental. Los intensificadores usualmente son sintagmas adverbiales que complementan verbos o adjetivos y fortalecen o debilitan su intensidad. De forma similar a los negadores, en los grafos de dependencias los intensificadores son subordinados por el nodo al que intensifican y pueden ser modelados de manera que afectan la polaridad total del árbol al que pertenecen, tomando en cuenta los árboles de raíz conjunción.

En la figura 6.7 se muestra un ejemplo de árbol de dependencias para la oración *«Luis es un muy buen jugador de fútbol.»* En el árbol se puede ver al nodo del adverbio *«muy»* subordinado por el nodo con el adjetivo *«buen»*. La magnitud de la intensificación sobre la polaridad sentimental del nodo padre se determina a partir de una lista de intensificadores, y se aplica sobre la polaridad sentimental del árbol, con el algoritmo de intensificación explicado en el módulo de sentimiento. En el ejemplo se muestra como el nodo intensificador *«muy»* tiene una intensificación de un 0,5 sobre la polaridad total del árbol, que se representa con una flecha violeta que va desde el nodo *«muy»* al nodo padre *«buen»* con una etiqueta *int: 0.5*.

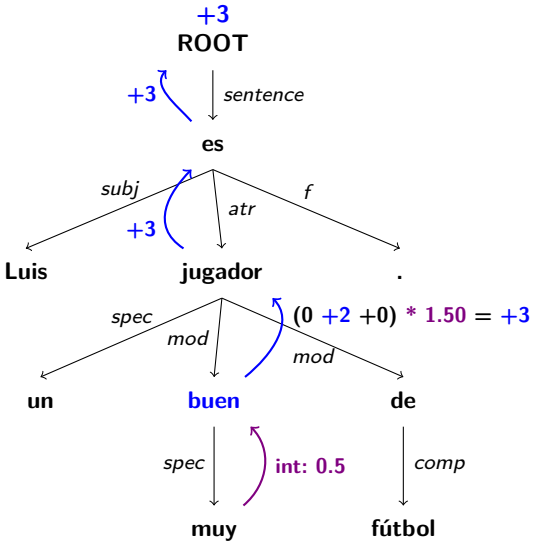

FIGURA 6.7: Ejecución ilustrativa del algoritmo de clasificación de oración con intensificación.

Otro fenómeno lingüístico que puede ser interpretado como intensificación son las oraciones adversativas. Las oraciones adversativas son introducidas por conectores adversativos como *pero*, *aunque*, *sin embargo*, que, como fue explicado en la sección 2.2, le dan una mayor importancia al contenido subordinado al conector. En este proyecto se decidió modelar este fenómeno como intensificación de la polaridad de los elementos subordinados al conector adversativo.

En la figura 6.8 se muestra el grafo de dependencias para la oración «*Luis es un excelente jugador pero su comportamiento en la cancha es irresponsable.*» que contiene una oración adversativa. Aplicando el algoritmo básico sobre el grafo, por un lado, la oración «*Luis es un excelente jugador*» tiene un resultado de polaridad sentimental de +4, mientras que la oración subordinada «*su comportamiento en la cancha es irresponsable*» tiene una polaridad sentimental de −4. Si sumamos ambas polaridades produce un resultado neutral, pero como están conectadas por un conector adversativo, se intensifica la polaridad de la oración adversativa. Una vez calculada la polaridad total del árbol de raíz *«pero»*, se la multiplica por dos, lo que lleva a que el resultado final de la oración completa sea −4, el cual se con dice más con la orientación semántica de la oración.

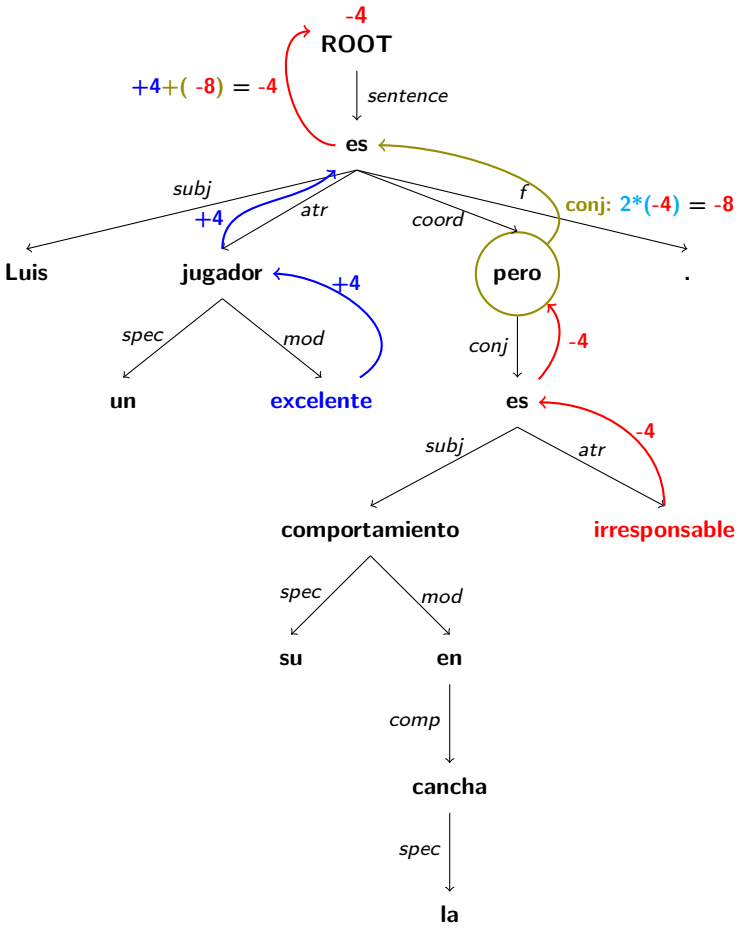

FIGURA 6.8: Ejecución ilustrativa del algoritmo de clasificación de oración con intensificación con oraciones adversativas.

En el algoritmo 6.5 se presenta en detalle la función que contempla tanto intensificadores como oraciones adversativas. Este parte del algoritmo básico, e introduce una variable intensificación. Para determinar la polaridad total de un árbol, se utiliza el módulo de sentimiento para analizar si los hijos que son *nodos hoja* pertenecen a la lista de intensificadores, y se obtiene la intensidad de la intensificación. Cuando existen múltiples intensificadores para un mismo árbol se realiza

una sumatoria de todos los valores de intensificación. Por último, de forma similar al algoritmo de negación, se aplica la intensificación sobre la polaridad total r calculada hasta el momento, se le suma la polaridad de los árboles de raíz conjunción r_conj, y se retorna el nuevo valor.

Algoritmo 6.5: Algoritmo para cálculo de polaridad de una oración con intensificación

```
Función clasificar_árbol(árbol)
    intensificación = 0
    r = 0
    r_conj = 0
    para hijo ∈ obtener_hijos(árbol) hacer
        si es_árbol(hijo) entonces
            si raiz_conj(hijo) entonces
                r_conj = clasificar_árbol(hijo) + r_conj
            en otro caso
                r = clasificar_árbol(hijo) + r
            fin
        en otro caso
            sent, neg, intensi = analizar_nodo(hijo)
            r = r + sent
            intensificación = intensificación + intensi
        fin
    fin
    sent, neg, intensi = analizar_nodo(obtener_raíz(árbol))
    r = r + sent
    intensificación = intensificación + intensi
    r = r · (1 + intensificación) + r_conj
    devolver r
end
```

6.3.2.4. Algoritmo completo

El algoritmo completo contempla tanto la negación como la intensificación. En la figura 6.9 se muestra el árbol de dependencias para la oración «Luis es un muy buen jugador de fútbol pero su comportamiento en la cancha no es el mejor.» que contiene tanto negación como intensificación. Aplicando de forma combinada los modelos de negación e intensificación presentados, se puede ver que el resultado obtenido es consistente con la orientación semántica de la oración.

En 6.6 se describe en detalle el algoritmo de clasificación completo. De forma análoga a los algoritmos 6.4 y 6.5 se analizan los hijos y la raíz de cada árbol, y se determinan las variables r, r_conj, *negación* e *intensificación*. Una vez que se tienen las variables se las combina para

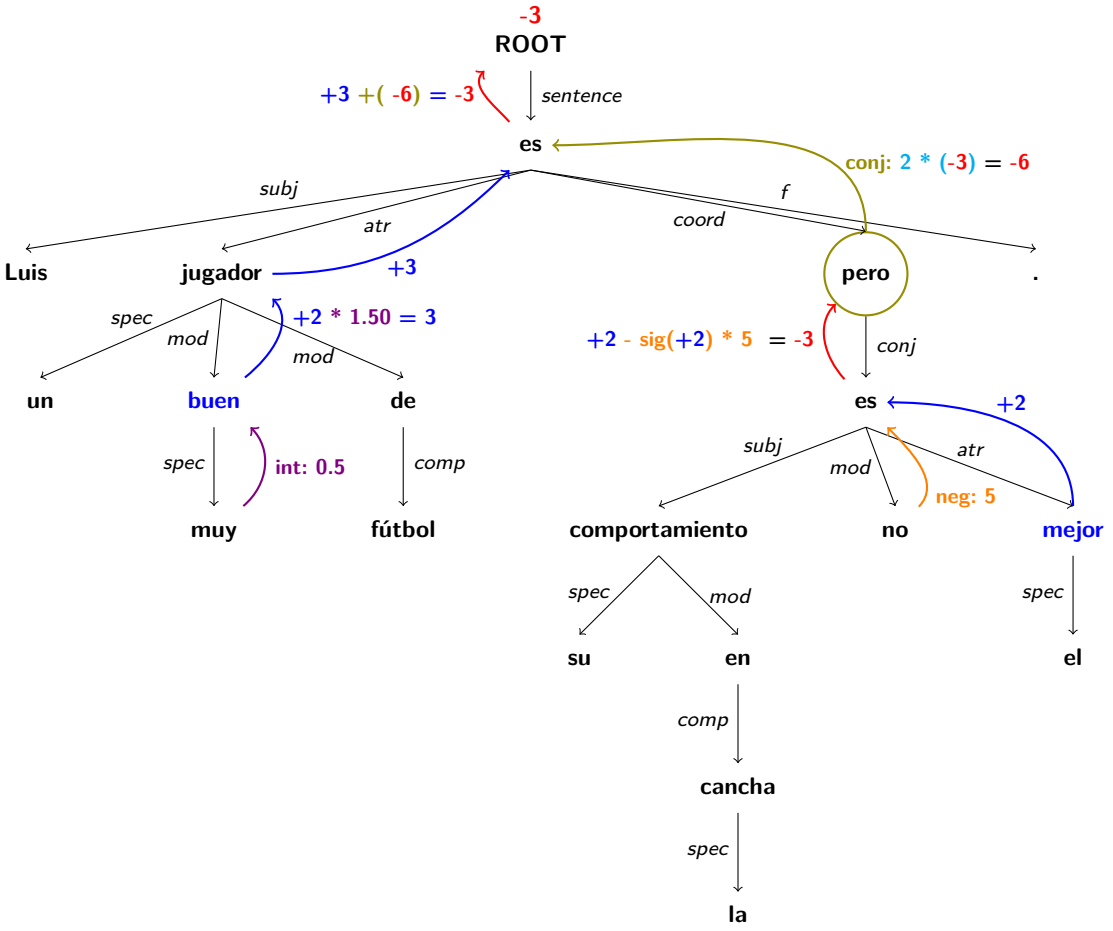

FIGURA 6.9: Ejecución ilustrativa del algoritmo de clasificación de oración con completo.

obtener el resultado final del árbol. Para combinarlas primero se intensifica el resultado, luego se le aplica la negación, y luego se le suma el resultado de los hijos árboles con raíz conjunción. Por último se retorna el resultado.

Algoritmo 6.6: Función para calcular la polaridad total de una oración

```
Función clasificar_árbol(árbol)
    negación = 0
    intensificación = 0
    r = 0
    r_conj = 0
    para hijo ∈ obtener_hijos(árbol) hacer
        si es_árbol(hijo) entonces
            si raiz_conj(hijo) entonces
                r_conj = clasificar_árbol(hijo) + r_conj
            en otro caso
                r = clasificar_árbol(hijo) + r
            fin
        en otro caso
            s, n, i = analizar_nodo(hijo)
            r = r + s
            si negación ≠ 0 entonces
                negación = −negación
            en otro caso
                negación = n
            fin
            intensificación = intensificación + i
        fin
    fin
    sent, neg, intensi = analizar_nodo(obtener_raíz(árbol))
    r = r + s
    negación = negación + n
    intensificación = intensificación + i
    r = r · (1 + intensificación)
    r = r + signo(r) · negación + r_conj
    devolver r
end
```

6.3.3. Modificación del lexicón sentimental

Luego de analizar el funcionamiento del clasificador sobre el conjunto de entrenamiento, se identificaron errores sobre opiniones con polaridad clara. Por ejemplo, la opinión «*La reunión de hoy me aburrió*» resultaba neutra. Al hacer un análisis de los errores, se identificó que la causa de este tipo de clasificaciones incorrectas eran provenientes del lexicón. Por un lado, existían palabras cuyo valor de sentimiento a priori era incorrecto, y por otro, el lexicón no incluía algunas palabras no neutras de uso frecuente en las opiniones, como es el caso de *'aburrir'* del ejemplo anterior.

Por esa razón se modifica el lexicón para lograr minimizar este tipo de errores. Para ello, se hace un estudio manual de las opiniones del corpus de entrenamiento que el clasificador etiquetó de forma errónea. En el estudio se determina qué palabras nuevas agregar al lexicón y

se modifican las polaridades a priori de algunas de ellas, así como sus intensidades de acuerdo al criterio de los anotadores. En total se agregaron 93 palabras nuevas y se cambió el sentimiento de otras 89.

Los cambios mejoran los resultados considerablemente en el corpus de entrenamiento, aunque por el método utilizado podría suponerse que se produjo un sobre-ajuste. Sin embargo, los resultados obtenidos sobre el corpus de test, muestran también una considerable mejora: se supera la linea base en más de 10 % tanto en accuracy como en F1-score promedio, como se puede ver en la tabla 7.2.3.

Algunos ejemplos de palabras que fueron agregadas al lexicón sentimental son:

- aburrir $\rightarrow -3$
- trabar $\rightarrow -3$
- encantar $\rightarrow 3$
- abuchear $\rightarrow -4$
- evadir $\rightarrow -3$
- favorecer $\rightarrow 3$
- legitimar $\rightarrow 2$
- censurar $\rightarrow -4$
- repeler $\rightarrow -3$
- repudiar $\rightarrow -3$
- aprobar $\rightarrow 2$
- cooperar $\rightarrow 2$
- inapropiar $\rightarrow -2$
- contrariar $\rightarrow -3$
- gustar $\rightarrow 2$

6.4. Clasificador basado en Aprendizaje Automático

En esta sección se especifica la implementación del clasificador basado en aprendizaje automático. Primero se describe brevemente el corpus de entrenamiento, luego la selección de features y preprocesamiento de los datos y finalmente los algoritmos utilizados.

Los resultados obtenidos para el clasificador automático se muestran en detalle en la sección 7.2.5.

6.4.1. Conjunto de entrenamiento y prueba

El aprendizaje automático requiere de un corpus de datos para aprender de ellos. Además, en este proyecto se utiliza aprendizaje supervisado por lo que los datos deben estar etiquetados como se explica en la sección 2.3.2.

Se cuenta con un corpus de prueba y uno de entrenamiento de similares características tanto en la composición como en la distribución de los datos. Se describen con detalle en el capítulo 5.

El corpus de prueba se utiliza únicamente para evaluar el resultado cuantitativo de los clasificadores y sus variantes. Para realizar evaluaciones cualitativas, para el desarrollo y mejora de las features, se utiliza siempre validación cruzada con cuatro particiones sobre el conjunto de entrenamiento. De lo contrario, se corre el riesgo de sobreajustar.

6.4.2. Preprocesamiento

Los métodos de aprendizaje automático requieren features numéricas, por lo que cuando se trabaja con textos se debe transformarlos en un vector numérico que los represente. Esta práctica se conoce como preprocesamiento.

Vectorización

Convertir los textos en vectores de features de tamaño fijo representando lo más importante de la información disponible es una de las principales tareas del aprendizaje automático aplicado al Procesamiento de Lenguaje Natural [22].

La forma más utilizada de hacerlo es *tokenizando* el texto y utilizando métodos basados en la bolsa de palabras.

Tokenizar un texto es transformarlo en una lista de componentes léxicos con significado. Un *token* representa una cadena de caracteres. Normalmente, cada símbolo ortográfico y cada palabra se consideran como un *token*, pero existen casos en los que no es así. Por ejemplo, para un nombre compuesto como *Juan Carlos* se le asigna un único token *Juan_Carlos*, mientras que para la palabra *al* se le asignan los tokens *a* y *el*.

A cada token de todos los que aparecen en el corpus de entrenamiento se lo agrega a un diccionario de tamaño **n**, siendo n la cantidad total de tokens.

Luego, el modelo de bolsa de palabras (en inglés *Bag of Words*), se trata de representar el texto como un vector que tiene como atributos cada uno de los *tokens* presentes en todo el corpus de entrenamiento. Lo hace sin tomar en cuenta el orden de aparición de los términos, y esa es la razón del nombre del modelo.

Existen varias alternativas para elegir los valores de cada feature, entre las que tres se destacan como las más comunes [2]:

- **Cuentas:** Se expresa la cantidad de veces que aparece el token en toda la opinión. Los valores posibles son números naturales (incluyendo el cero).
- **Presencia:** Se expresa en un booleano la presencia o no de el token en toda la opinión. Los posibles valores posibles son cero o uno. Suele obtener mejores resultados que el de cuentas para análisis de sentimiento [21].
- **TF-IDF:** Se expresa cuán relevante es el token en el texto con respecto a su presencia en el corpus de entrenamiento. El valor surge de una división entre un valor de frecuencia del término en el ejemplo sobre un valor que representa lo frecuente que es en todos los textos del corpus. Los posibles valores son los reales no negativos.

Para mejorar los resultados se utilizan, en lugar de las palabras, sus lemas. De esta forma, palabras similares como verbos en sus distintas conjugaciones o plurales son considerados como el mismo *token*. Por lo tanto es más correcto utilizar el término *bolsa de lemas*, aunque se acostumbra llamarlo *bolsa de palabras* de todas formas. Los lemas utilizados son los que otorga Freeling, herramienta detallada en la sección 6.1.

El preprocesamiento, incluidas las modificaciones que se describen a continuación, son implementadas con la librería Scikit-learn.

N-Gramas

Los n-gramas son features que representan n tokens que aparecen juntos en el texto. Por ejemplo, en la frase «un día hermoso», los unigramas son *un*, *día* y *hermoso*; los bigramas *un día* y *día hermoso*; y el único trigrama es *un día hermoso*.

Es común agregar bigramas o trigramas a las features en la vectorización en el análisis de sentimiento [21] ya que, pese a que aumentan la dimensionalidad, permiten representar expresiones como *muy bueno* o *no tan malo* dando mayor poder de expresividad.

Mínima frecuencia de documento

Se prueba la opción de no tomar en cuenta los *tokens* que aparecen únicamente en un ejemplo (a veces llamados *documentos* y de ahí el nombre) en todo el corpus, ya que su peso estadístico no es suficiente para aportar información sobre la relación con las posibles clases. En ese caso se considera el MIN-DF (por su nombre en inglés *Minimum Document Frequency*) como 2, ya que el token debe estar en al menos dos documentos. Que MIN-DF sea 1 significa tomar en cuenta todas las palabras.

Stopwords

Se denomina *stopwords* o *palabras vacías* a ciertas palabras como artículos, pronombres, preposiciones y otras que son muy comunes en el idioma pero carecen de significado semántico. Estas palabras suelen agregar ruido a los modelos por lo que se decide quitarlas.

En este proyecto se utilizan tres opciones con respecto a las stopwords:

- **Ninguna:** No se quita ninguna palabra del vocabulario.
- **Propias:** Se quitan los nombres de las fuentes seleccionadas para recuperar las opiniones de entrenamiento. Se notó durante el proyecto que muchos de los nombres de las fuentes quedaban fuertemente correlacionados con una categoría, generando que opiniones cortas y muchas veces carentes de polaridad fueran mal clasificadas. Por ejemplo, se clasificaba «Batlle dijo que lo pensaría» como negativa, debido a que el token *Batlle* se relacionaba a la clase negativa. Por esta razón, se agregan como stopwords.
- **Todas:** Además de las fuentes, se quitan las palabras pertenecientes a la lista de stopwords de NLTK para el español.

6.4.3. Selección de features

Además de las features generadas en el preprocesamiento, se busca ampliar la información que describe a cada texto y que recibe el algoritmo para mejorar la clasificación.

Se agregan entonces features con cuentas de palabras con polaridad presentes el los lexicones sentimentales que fueron descritos en la sección 6.1. El método consiste en contar la cantidad de ocurrencias de adjetivos, verbos, adverbios y sustantivos, diferenciando los positivos de los negativos; por lo tanto se obtienen dos features por cada una de las cuatro categorías gramaticales nombradas.

Por ejemplo, la opinión *«Recordó que la sociedad uruguaya ha sido benévola con los servidores públicos y casi cruel con los demás trabajadores»* tiene una ocurrencia de adjetivos positivos (*benévola*) y una ocurrencia de adjetivos negativos (*cruel*).

Luego de agregadas estas features, se aumenta muy levemente la dimensionalidad del espacio de búsqueda, como se puede ver en la tabla 6.4. Sin embargo, los resultados mejoran notoriamente como se puede apreciar en la sección 7.2.5.

Las features de lemas, como se explica en la sección 6.4.2, difieren en sus valores según el tipo de preprocesamiento, pero siempre se mantiene su cantidad.

Feature	Cantidad
Lemas	1437
Cuentas de palabras con polaridad	8
Total	**1445**

CUADRO 6.4: Cantidad de features en el clasificador de aprendizaje automático

Es importante tener en cuenta que no siempre el agregado de features mejora el rendimiento de los algoritmos de aprendizaje automático. El aumento de la dimensionalidad generado y el posible ruido agregado por una feature mal elegida puede generar sobreajuste y por lo tanto empeorar los resultados.

Por esta razón se mantienen en la vectorización de este trabajo únicamente las features que generan una mejora en el resultado. Fueron descartadas en el proceso de selección múltiples candidatos tales como el largo en tokens de las opiniones, la fuente de las mismas, la suma de las polaridades de las palabras de la oración presentes en el lexicón sentimental, etc.

Transformación de los datos

Es normal que algunas de las features tomen valores en rangos más amplios que otras. Esto genera que algunas features, las de mayor rango, pesen más en la decisión de algunos algoritmos. Para evitarlo, es común transformar los datos manteniendo la proporción entre ellos.

En el caso de la utilización de clasificadores basados en Naive Bayes, se debe reescalar los datos. Esto significa transformar los datos para que queden proporcionalmente distribuidos en un mismo rango, en este caso el rango $[0, 1]$.

Además de lo anteriormente explicado, Multinomial Naive Bayes no admite valores negativos, por lo que el reescalado se hace indispensable.

En el caso de la utilización de clasificadores basados en SVM, se realiza una reducción de dimensionalidad con Análisis de Componentes Principales o PCA (por sus siglas en inglés).

PCA es una técnica que utiliza la descomposición SVD para generar una nueva base de coordenadas con los vectores propios de la matriz de covarianza de los datos. De esta forma, se pueden proyectar los datos en un espacio de menor dimensionalidad, pero manteniendo las dimensiones que más aportan en su varianza y por lo tanto las que más ayudan en la clasificación.

Peso de las clases

Los algoritmos asignan mayor peso a las clases que tienen mayor cantidad de ejemplos en el conjunto de entrenamiento. En Naive Bayes, por ejemplo, la proporción del tamaño de cada clase es utilizada como probabilidad a priori de pertenencia a esa clase.

En el conjunto de entrenamiento predominan, aunque con poca diferencia, los ejemplos negativos, como se puede ver en el cuadro 5.1. Sin embargo, es común asumir en el análisis de sentimiento que todas las clases tienen la misma probabilidad a priori de ser correctas para un ejemplo dado.

Por otro lado, durante el proceso de análisis de los resultados se notó que el recall (las métricas se explican en la sección 5.2.1.1) de la clase Neutral es lo que genera más problemas.

Por esta razón se decide explorar tres posibilidades:

- **Por defecto:** se mantienen las probabilidades del conjunto de entrenamiento.
- **Balanceadas:** se explicita que las tres clases tienen un tercio de la probabilidad a priori.
- **Más peso al neutral:** se le da la mitad de la probabilidad al neutral y un cuarto a cada una de las restantes. De esta forma se intenta mejorar el recall de la clase Neutral.

6.4.4. Algoritmos de clasificación

En esta sección se presentan los algoritmos de aprendizaje supervisado elegidos en este proyecto: Support Vector Machines (SVM) y Multinomial Naïve Bayes (MNB). Luego, se describe brevemente la implementación y los parámetros elegidos para cada uno de ellos.

Se trata de dos de los métodos más utilizados en el área del análisis de sentimiento en texto. Ambos algoritmos pertenecen al aprendizaje automático supervisado, que es descrito en la sección 2.3.2.

Se probaron en el proyecto otros algoritmos tales como algunos basados en árboles de decisión, pero fueron descartados debido a que no generaron buenos resultados.

Multinomial Naïve Bayes

Es un modelo probabilístico, que predice la respuesta correcta modelando el problema como una distribución de probabilidad. Es un modelo simple pero de todas formas performante, por lo que se utiliza con frecuencia [18].

La base del modelo es el teorema de Bayes, que permite calcular la probabilidad condicional de un evento A sujeto a la ocurrencia de otro evento B,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Es llamado Naïve porque se basa también en fuertes suposiciones de independencia condicional entre las features. Esta independencia normalmente no se corresponde con la realidad, pero el algoritmo funciona correctamente de todas formas.

Si se llama c al atributo objetivo con sus valores posibles y siendo A_i el valor de cada atributo, se busca un c tal que:

$$\begin{aligned} c &= \arg\max_{c \in C} P(c|a_1, \ldots, a_n) \\ &= \arg\max_{c \in C} \frac{P(a_1, \ldots, a_n|c)P(c)}{P(a_1, \ldots, a_n)} \\ &= \arg\max_{c \in C} P(a_1, \ldots, a_n|c)P(c) \end{aligned}$$

Aquí entra en juego la suposición de probabilidad condicional, que permite pasar de la fórmula anterior a:

$$c = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|c)$$

Por lo tanto, se elige como clase a la que maximiza la probabilidad de ser la correcta.

La probabilidad a priori de cada clase P(c) se puede estimar de los datos o tomar como valor de entrada del algoritmo. Ambas técnicas fueron usados en este proyecto, como se explica en la sección 6.4.3.

El algoritmo asume un modelo de eventos con una distribución de probabilidad. Es común utilizar la distribución normal para los problemas de naturaleza continua. Sin embargo, el modelado de problemas de texto es más apropiado a distribuciones discretas. En particular se modela con la distribución multinomial: por esa razón se lo denomina Multinomial Naive Bayes.

Support Vector Machines

SVM, por sus siglas, es un algoritmo que genera predictores lineales en espacios de alta dimensionalidad [27].

El modelo construye los hiperplanos con mayor separación a las instancias más cercanas de cada una de las clases. Este sencillo fundamento es sin embargo poderoso y puede llevar a complejidades derivadas de la alta dimensionalidad.

Es común que las clases no sean en realidad linealmente separables. Por esa razón, es frecuente aplicar transformaciones a los datos que aumentan la dimensionalidad de los mismos para poder ahí separarlos. La función de transformación es comúnmente llamada *kernel* [27].

6.4.5. Implementación y parámetros

Los algoritmos fueron utilizados con la implementación brindada por la librería Scikit-learn.

Ambos métodos tienen parámetros que fueron ajustados y probados en el proyecto mediante un gridsearch como es explicado en 7.2.4.

Algunos de los parámetros se describen a continuación:

En el clasificador MNB, además de los pesos de las clases, se modifica el parámetro ***alpha*** con los valores 0, 0.33, 0.66 y 1. *Alpha* es un parámetro de suavizado como se explica en la sección 2.3.2.

En el clasificador de SVM se usan los **kernel** lineal, RBF y polinómico de grado 2 y 3. Además, se utiliza el parámetro ***C*** en 0.01, 1 y 100 y ***gamma*** en automático, 1 y 0.01. Para una descripción de estos atributos, se puede ver la sección 2.3.2.

6.5. Clasificador híbrido

Luego de creados los clasificadores basados en reglas y aprendizaje automático, se construye un clasificador híbrido que combina las dos metodologías. La premisa para la creación de este nuevo clasificador es utilizar las reglas como base de nuevas posibles features para una mejora del clasificador de aprendizaje automático [26].

En esta sección se detallan las features agregadas y las modificaciones realizadas a las existentes para agregar datos de importancia obtenidos del procesamiento del clasificador basado en reglas.

Analizando los resultados del clasificador de aprendizaje automático, se concluye que el algoritmo SVM se adapta mejor a este problema que MNB, como se puede ver en la sección de resultados 7.2.5. Por lo tanto se utiliza solamente SVM en la creación del clasificador híbrido.

Se agregan dos clases de features: la que refleja el resultado final del algoritmo de reglas y las que aportan la información obtenida sobre cada palabra.

6.5.1. Feature de resultado de la clasificación de reglas

La nueva feature es el resultado obtenido en la clasificación basada en reglas. Como se detalla en la sección 6.3 el resultado es un número que señala el grado de positividad o negatividad de la opinión general. Ese número se agrega como feature al clasificador.

6.5.2. Extensión de bolsa de lemas

El principal problema del modelo bolsa de palabras es que deja fuera de la representación las relaciones sintácticas y semánticas que existen entre las palabras.

Se explora por esta razón la modificación o creación de features que reflejen no solamente la presencia de un token o lema en la opinión, sino además un valor que indique si la palabra está presente y si se encuentra negada o intensificada en el contexto de la opinión.

En el análisis de los resultados del clasificador de aprendizaje automático, se vio que la diferencia entre los vectorizadores de cuentas y de presencia es insignificante. Esto se debe a que al ser cortas las opiniones normalmente las palabras no se utilizan más de una vez en cada una. Este comportamiento del algoritmo es normal, como se explicó en la sección 6.4.2. Por lo tanto se decide prescindir de la información de la cuenta de veces que está presente cada lema.

Primera versión de la extensión de bolsa de lemas

Se asigna un puntaje a cada lema que refleja, no la polaridad, sino la modificación a la polaridad original de esa palabra por el contexto de la oración.

Este puntaje es negativo si la aparición de la palabra se da en un contexto de negación y positivo de lo contrario. El proceso con el que se modela la negación se describe en detalle en la sección 6.3.2.2.

Además, el puntaje aumenta o disminuye su valor absoluto en función del contexto de intensificación en el que se encuentra. El modelado del alcance de la intensificación se describe en detalle en la sección 6.3.2.3.

Si el lema no está presente en la opinión el puntaje es cero.

Entonces, en la oración «Es una muy buena persona y no molesta»:

- El lema ***ser*** tiene puntaje 1.
- El lema ***buen*** tiene puntaje 1.5 debido a que está intensificado.
- El lema ***molestar*** tiene puntaje -1 porque está negado.
- Cualquier otro lema que no esté en la oración tiene puntaje 0.

El resultado de esta primera versión no fue positivo, obteniendo valores similares en todas las métricas. Esto se debe a que el algoritmo puede confundir la información sobre la presencia o no de una palabra con el contexto en el que está, al ser el valor correspondiente a la no presencia el medio del posible rango de valores. Esta dificultad se hace más importante porque

los vectores son dispersos (la mayoría de los valores son cero), ya que están presentes apenas unas decenas de palabras de las cientas que existen en el diccionario.

Segunda versión de la extensión de bolsa de lemas

Con el fin de resolver la confusión descrita en la primera versión, se decide generar dos puntajes por lema en lugar de uno.

El primero de ellos refleja la presencia del lema en un contexto ausente de negación. El segundo refleja si está presente en un contexto de negación.

En ambos casos se utiliza signo positivo y se toma en cuenta la intensificación de la misma forma que antes.

Nuevamente en el ejemplo «Es una muy buena persona y no molesta»:

- El lema ***ser*** tiene puntajes (1,0) porque la palabra ser está en contexto de no negación y no está en contexto de negación.
- El lema ***buen*** tiene puntajes (1.5,0) debido a que está en contexto de no negación e intensificado allí.
- El lema ***molestar*** tiene puntajes (0,1) porque está presente únicamente en contexto de negación.
- Cualquier otro lema que no esté en la oración tiene puntajes (0,0).

De esta forma la no presencia de las palabras se representa con el mínimo del rango posible en ambos valores.

Por lo tanto, se duplica la cantidad de features generadas por la vectorización, lo que genera un impacto negativo por el aumento de la dimensionalidad como se puede ver en la tabla 6.5. Sin embargo, ese efecto es notoriamente menor que el impacto positivo de agregar esta información, como se podrá ver en los resultados obtenidos en la sección 7.2.6.

Feature	Cantidad
Lemas no negados	1437
Lemas negados	1437
Cuentas de palabras con polaridad	8
Resultado clasificador reglas	1
Total	**2883**

Cuadro 6.5: Descripción cuantitativa de features del sistema

Capítulo 7

Resultados

La evaluación de los métodos automáticos de Procesamiento de Lenguaje Natural no es sencilla. Esto se debe principalmente al riesgo que existe de sobreajustar y a que en muchos casos, entre los que se encuentra el análisis de sentimiento, es imposible lograr una eficacia del 100 por ciento por lo que hay que ser cuidadoso al definir las líneas base y tope y analizar los resultados.

7.1. Metodología

En esta sección se describe el procedimiento para realizar la evaluación de los resultados. Primero se describe brevemente el conjunto de prueba y luego las métricas elegidas y el por qué de la elección de cada una.

7.1.1. Corpus de prueba

Todas las pruebas descritas en este capítulo fueron ejecutadas sobre un corpus de prueba formado por 938 opiniones. Este conjunto fue utilizado únicamente para estos fines de forma de evitar el sobreajuste. Los detalles de la creación y características de este conjunto fueron descritos en el capítulo 4.

7.1.2. Métricas

Es necesario definir medidas que permitan evaluar de forma objetiva el desempeño de un clasificador creado. Existen varias métricas que permiten en su conjunto conocer su desempeño y compararlo con otros clasificadores similares, aunque estos resultados dependen también del corpus que se utiliza. De todas ellas, accuracy, precision, recall y F-1 score son un estándar en

el campo del Aprendizaje Automático y el Procesamiento de Lenguaje Natural y son los que se usan en este proyecto. Las métricas fueron descritas en la sección 5.2.1.1.

7.2. Resultados

En esta sección se describen los resultados obtenidos por el clasificador para los algoritmos generados en este proyecto. Previamente se exponen los resultados de las líneas base y tope, de forma de poder evaluarlos adecuadamente.

7.2.1. Línea base

De las tres líneas base propuestas, explicadas en detalle en la sección 6.2, la más compleja y que mejores resultados genera es la sumatoria de polaridades. Se construye mediante la suma de los valores de las palabras que están en la opinión y pertenecen a una lista de palabras positivas y negativas.

Los resultados de la sumatoria de polaridades son los siguientes:

	Precision	Recall	F1-score
Negativo	0.50	0.48	0.49
Neutral	0.44	0.29	0.35
Positivo	**0.52**	**0.66**	**0.58**
Promedio	0.48	0.49	0.48
Accuracy total: **0.50**			

CUADRO 7.1: Resultados de la línea base

7.2.2. Línea tope

El límite superior de resultados posibles está marcado por el análisis de concordancia entre los anotadores de los datos. Como fue explicado anteriormente, el análisis de sentimiento de opiniones de prensa, en su mayoría políticas, es complejo incluso para los humanos. Al generarse los resultados comparando la salida del programa con anotaciones de humanos, se puede ver que si los humanos no concuerdan entre ellos, es imposible que el programa coincida con todos ellos al mismo tiempo, lo que hace imposible llegar al 100 por ciento de accuracy. Por lo tanto, la correctitud del algoritmo puede ser tan alta como la concordancia entre los anotadores y es contra esa cifra que debe realizarse una comparación. Estos resultados se pueden observar al detalle en la sección 5.2.1. Incluso, se puede decir que en caso de llegar a ese punto, el programa es tan bueno haciendo juicios de sentimiento como cualquiera de los

anotadores humanos. Mejores resultados que éste reflejarían sobreajuste al corpus de testeo o a la subjetividad de uno de los anotadores.

El análisis de concordancia de este proyecto, explicado en la sección 5.2, muestra que el tope de posibles resultados para este proyecto es de:

	Precision	Recall	F1-score
Promedio	0.74	0.72	0.72
Accuracy total: **0.72**			

Cuadro 7.2: Resultados entre A2 y A3

7.2.3. Resultados del sistema de reglas

Como se detalla en la sección 6.3.2, el clasificador basado en reglas tiene variantes. Por un lado se tiene el algoritmo básico que suma los valores de las palabras de cada nodo del árbol de dependencias, calculando así el resultado de la opinión. Además se cuenta con un algoritmo de negación que agrega al análisis el efecto de los negadores en las opiniones; y por último el algoritmo de intensificación que agrega la análisis del efecto de los intensificadores.

Se presentan a continuación los resultados para las cuatro combinaciones, utilizando o dejando de lado cada una de las variantes, tanto para el corpus de prueba en el cuadro 7.3, como para el de entrenamiento en el cuadro 7.4.

Algoritmo			
Con intensificación	Con negación	Accuracy (%)	F1-score (%)
No	No	64.5	65
No	Sí	66.0	**67**
Sí	No	65.1	66
Sí	Sí	**66.5**	**67**

Cuadro 7.3: Resultados del sistema de reglas en corpus de entrenamiento

Algoritmo			
Con intensificación	Con negación	Accuracy (%)	F1-score (%)
No	No	60.2	60
No	Sí	**62.3**	**62**
Sí	No	59.9	60
Sí	Sí	62.0	**62**

Cuadro 7.4: Resultados del sistema de reglas en corpus de prueba

Los sistemas de reglas, a diferencia de los de aprendizaje automático, no requieren entrenamiento con un corpus de datos. Por esta razón, se pueden evaluar tanto en el conjunto de prueba como en el de entrenamiento sin que exista diferencia a priori.

Sin embargo, el corpus de entrenamiento y los resultados en él se utilizaron para corregir errores y mejorar las listas de palabras, como se explica en la sección 6.3.3. Por lo tanto existe un sobreajuste al conjunto de entrenamiento que explica los mejores resultados en él.

Por otro lado, el accuracy en el conjunto de prueba es ligeramente mejor sin utilizar el algoritmo de intensificación. Sin embargo, se elige el clasificador que lo contiene como el final ya que la decisión se basa en los resultados obtenidos en el conjunto de entrenamiento (de modo de no sobreajustar). Además, un 0.3 % menos de accuracy en el conjunto de prueba representa apenas 3 opiniones, por lo que la diferencia puede considerarse insignificante.

Por lo tanto, se elige el algoritmo con intensificación y negación como el algoritmo final basado en reglas, y sus resultados completos se muestran en el cuadro 7.5 y su matriz de confusión en el cuadro 7.6.

	Precision	Recall	F1-score
Negativo	0.66	0.56	0.60
Neutral	0.53	0.58	0.55
Positivo	**0.67**	**0.69**	**0.68**
Promedio	0.62	0.62	0.62
Accuracy total: **0.62**			

Cuadro 7.5: Resultados completos del clasificador de reglas

	Negativo	Neutral	Positivo
Negativo	**144**	64	51
Neutral	38	**164**	82
Positivo	37	84	**274**

Cuadro 7.6: Matriz de confusión del clasificador de reglas

Estos resultados mejoran sensiblemente los conseguidos en la línea base. Esto se debe a que, como fue explicado a lo largo de este informe, las relaciones entre las palabras son importantes para detectar sus significados y este sistema los tiene en cuenta, al contrario de la línea base.

7.2.4. Búsqueda de grilla

Para los métodos basados en aprendizaje automático, como fue explicado en la sección 6.4, se utilizaron varios métodos de preprocesamiento y dos algoritmos de clasificación, contando cada uno de ellos con algunos parámetros ajustables.

Es común que los diferentes parámetros que tienen los algoritmos se ajusten mejor a unos u otros tipos de datos y de preprocesamiento. Sin embargo, es difícil saber a priori cuál será el más adecuado para un conjunto de datos en particular.

Por esa razón, con el fin de descubrir y seleccionar la mejor combinación de algoritmo de preprocesamiento y clasificación, así como los parámetros de cada uno de ellos, se prueba con cada una de las combinaciones posibles. Luego se analizan los resultados obtenidos y se selecciona la combinación que genera los mejores. Este proceso es conocido como *búsqueda de grilla* o *gridsearch*.

Los parámetros de la búsqueda fueron tanto para el preprocesamiento como para los algoritmos de aprendizaje automático. Los elegidos para preprocesar el texto serán explicados en las siguientes secciones 7.2.5 y 7.2.6 junto con los resultados generados. Los parámetros de los clasificadores incluidos en el gridsearch no fueron distinguidos en cuanto a resultados obtenidos, sino que se muestra el mejor. Estos parámetros son descritos en la sección 6.4.4.

De las 7632 combinaciones, un 38 % generaron menos de 40 % de accuracy, 15 % generaron un accuracy de entre 40 y 50 %, 42 % entre 50 y 60 % y el restante 5 % un accuracy mayor a 60 %.

7.2.5. Resultados aprendizaje automático

Debido a la utilización de gridsearch, generando más de cinco mil resultados, se hace imposible mostrar todos en este documento. Por esta razón se presenta para cada parámetro de preprocesamiento el mejor resultado obtenido para cada una de sus posibles opciones.

Los resultados fueron obtenidos mediante una validación cruzada de 4 particiones y son los presentados en el cuadro 7.7. Luego, cada una de las combinaciones elegidas se entrenó con todo el conjunto de entrenamiento y se probaron en el conjunto de prueba; estos resultados son presentados en el cuadro 7.8.

		SVM		MNB	
Parámetro	Valor	Accuracy	F1-score	Accuracy	F1-score
Mínima frecuencia de documento	1	**65.7**	**66**	**57.1**	56
	2	65.1	65	56.6	**57**
Stopwords	Ninguna	65.5	**66**	56.8	**57**
	Propias	65.3	65	56.7	**57**
	Todas	**65.7**	**66**	**57.1**	56
Vectorizador	TF-IDF	**65.7**	**66**	56.4	**56**
	Cuenta	64.6	65	**57.1**	**56**
Rango de n-gramas	1	65.1	65	**57.1**	**56**
	2	65.5	**66**	55.5	55
	3	**65.7**	**66**	55.3	55
Features agregadas	Ninguna	54.1	54	54.5	54
	Cuentas	**65.7**	**66**	**57.1**	**56**

Cuadro 7.7: Resultados del aprendizaje automático en corpus de entrenamiento

		SVM		MNB	
Parámetro	Valor	Accuracy	F1-score	Accuracy	F1-score
Mínima frecuencia de documento	1	60.8	**61**	53.4	52
	2	**60.9**	61	**54.5**	**54**
Stopwords	Ninguna	60.2	60	**54.5**	**54**
	Propias	60.2	60	54.1	**54**
	Todas	**60.9**	**61**	53.2	52
Vectorizador	TF-IDF	60.2	60	54.2	52
	Cuenta	**60.9**	**61**	**54.5**	**54**
Rango de n-gramas	1	**60.9**	**61**	**54.5**	**54**
	2	60.0	60	53.4	52
	3	60.0	60	53.2	52
Features agregadas	Ninguna	52.9	51	51.7	50
	Cuentas	**60.9**	**61**	**54.5**	**54**

Cuadro 7.8: Resultados del aprendizaje automático en corpus de prueba

Como fue explicado en la sección 6.4, cuando el clasificador utilizado es SVM se preprocesan los datos con PCA. Existe la posibilidad de elegir la cantidad de componentes y se probó tanto con dos mil como con cuatro mil. Todas las combinaciones probadas dieron exactamente los mismos resultados con cualquiera de las opciones por lo que no se presentan en las tablas de resultados.

En el análisis de los resultados se puede notar que SVM se ajusta mejor a la resolución del este problema, generando mejores resultados que MNB en todas las combinaciones.

Se puede ver además que el clasificador mejora notoriamente agregando las features de cuentas de palabras.

Por otra parte, el resto de los parámetros no generaron diferencias mayores al 2 % en su mejor combinación. Estas diferencias son pequeñas y teniendo en cuenta la naturaleza probabilística de los métodos, podrían incluso considerarse insignificantes.

Sin embargo, se ve que los bigramas y trigramas no mejoraron los resultados, sino que incluso los bajaron levemente. Este tipo de features suelen mejorar el rendimiento en el análisis de sentimiento ya que permiten capturar expresiones, negaciones, intensificaciones y otros tipos aspectos del lenguaje formados por más de una palabra. No fue este el caso, debido seguramente a que el agregado bigramas y trigramas, por otra parte, aumenta considerablemente el espacio dimensional de la búsqueda. Además, estas nuevas features son de menor probabilidad de aparición: es menos probable encontrar un bigrama que cualquiera de las dos palabras que lo conforman. Este espacio con más dimensiones de menores probabilidades requiere un conjunto de entrenamiento más grande, algo con lo que no se cuenta en este proyecto.

La mejor combinación de parámetros define el clasificador final construido con aprendizaje automático. Se presentan por lo tanto sus resultados completos en el cuadro 7.9 y su matriz de confusión en el cuadro 7.10.

	Precision	Recall	F1-score
Negativo	0.58	0.60	0.59
Neutral	0.57	0.55	0.56
Positivo	**0.65**	**0.66**	**0.65**
Promedio	0.60	0.60	0.60
Accuracy total: **0.61**			

CUADRO 7.9: Resultados completos del clasificador de aprendizaje automático

	Negativo	Neutral	Positivo
Negativo	**157**	46	57
Neutral	48	**155**	80
Positivo	64	72	**259**

CUADRO 7.10: Matriz de confusión del clasificador de aprendizaje automático

Los resultados obtenidos por el clasificador de aprendizaje automático son similares a los conseguidos por el basado en reglas. Ambos mejoran sensiblemente la línea base, pero aún no se acercan a la línea tope.

La principal razón por la que no se ven mejores resultados es el tamaño del corpus de entrenamiento, que no llega a ser lo suficientemente representativo para que el algoritmo pueda generalizar correctamente muchos de los casos. Además, el hecho de que la vectorización con TF-IDF, que usualmente mejora los resultados, prediga prácticamente con la misma precisión que la vectorización de cuentas, indica nuevamente que los ejemplos disponibles no son suficientes.

Para solucionar este problema puede haber dos alternativas: agrandar el corpus de entrenamiento o mejorar la información que se puede obtener de cada uno de los ejemplos, que es lo que motiva la cración del clasificador híbrido.

7.2.6. Resultados del clasificador híbrido

El clasificador híbrido fue construido mediante la transformación de las cuentas de palabras, duplicando las features correspondientes a cada token, y el agregado como feature del resultado del sistema de reglas. Esto se explica en detalle en la sección 6.5.

Nuevamente se obtienen los resultados utilizando gridsearch y con validación cruzada de 4 particiones. En el cuadro 7.11 se presenta el mejor resultado de todas las combinaciones de parámetros antes y después de aplicar cada uno de los cambios de features. De la misma

forma que en la sección anterior, se muestra también el resultado de cada una de las combinaciones elegidas en el conjunto de prueba, luego de entrenar con la totalidad del conjunto de entrenamiento. Los resultados se muestran en el cuadro 7.12.

		SVM	
Parámetro	Valor	Accuracy (%)	F1-score (%)
Vectorizador	Cuentas	66.3	66
	1ra versión de la extensión	66.6	66
	2da versión de la extensión	**69.7**	**70**
Features agregadas	Ninguna	65.7	66
	Resultado de reglas	**69.7**	**70**

Cuadro 7.11: Resultados del clasificador híbrido en corpus de entrenamiento

		SVM	
Parámetro	Valor	Accuracy (%)	F1-score (%)
Vectorizador	Cuenta	62.6	63
	1ra versión de la extensión	62.8	62
	2da versión de la extensión	**64.3**	**64**
Features agregadas	Ninguna	60.9	61
	Resultado de reglas	**64.3**	**64**

Cuadro 7.12: Resultados del clasificador híbrido en corpus de prueba

	Precision	Recall	F1-score
Negativo	0.59	**0.69**	0.64
Neutral	0.59	0.58	0.58
Positivo	**0.72**	0.66	**0.69**
Promedio	0.65	0.64	0.64
Accuracy total: **0.64**			

Cuadro 7.13: Resultados completos del clasificador híbrido

	Negativo	Neutral	Positivo
Negativo	**178**	46	35
Neutral	55	**164**	65
Positivo	67	67	**261**

Cuadro 7.14: Matriz de confusión del clasificador híbrido

Los resultados mejoran notoriamente con las dos modificaciones, llegando a un 64 % tanto en accuracy como en F1-score promedio. En particular, el recall de los neutros, la métrica que peores resultados genera, mejora hasta llegar al 58 % logrado con el sistema de reglas. Por otra parte la categoría de los positivos es la que logra mejores valores, llegando a un 69 % en F1-score.

7.2.7. Resultados con tolerancia

Muchas de las opiniones del corpus pueden ser catalogadas como pertenecientes a más de una de las tres categorías posibles. En esos casos, se anota en el corpus una segunda opción además de la categoría elegida inicialmente.

> *«Tenemos claro que esto es circunstancial. Si bien es muy grato, no influye a la hora de elaborar una idea artística, un espectáculo o grabar un disco»* (7.1)
>
> Tabaré

Por ejemplo, la opinión 7.1 fue clasificada como *Neutral*. Sin embargo, tiene un matiz positivo («es muy grato») por lo que sería excesivo catalogar como errónea una clasificación de *Positivo* en este caso y se agrega como segunda opción.

El total de opiniones con una segunda opción en el conjunto de test es de 126, representando un 13.4 % del total.

Se realiza entonces una evaluación más flexible en la que, en los casos en que una opinión tiene dos posibles resultados, se considera cualquiera de ellos como correcto. Los resultados obtenidos con esta evaluación *tolerante* se presentan en los cuadros 7.15 y 7.16.

	Precision	Recall	F1-score
Negativo	0.63	**0.73**	0.67
Neutral	0.66	0.67	0.67
Positivo	**0.78**	0.70	**0.74**
Promedio	0.69	0.70	0.69
Accuracy total: **0.70**			

Cuadro 7.15: Resultados finales con tolerancia

	Negativo	Neutral	Positivo
Negativo	**188**	36	34
Neutral	47	**184**	45
Positivo	65	57	**282**

Cuadro 7.16: Matriz de confusión con tolerancia

Es importante en este caso comparar la mejora obtenida con la posible mejora de haber simplemente agregado una clase aleatoria a 126 opiniones aleatorias. Considerando que 55 de las 126 ya habían sido correctamente clasificadas, la oportunidad de mejora es de 76 opiniones, un 8.1 %. Por lo tanto, el resultado de mejora en accuracy es de un 5.3 % en un 8.1 %: un 65 %

de lo posible. Esto confirma que la mejora es mayor que si se hubiera agregado una clase aleatoria, un 50 % del total de lo posible: 4.05 %.

Estos resultados son muy similares a los de la línea tope, aunque no sería correcto compararlos ya que la forma de evaluación es distinta (no se permite más de una respuesta correcta). Sin embargo, se reafirma que existe una cantidad significativa de opiniones cuya polaridad es dudosa entre dos clases.

7.2.8. Resultados finales

En esta sección se resumen los resultados principales de cada uno de los tres clasificadores contraídos. No se presentan nuevos resultados pero sí se presentan juntos en la el cuadro 7.17 de forma de que puedan ser fácilmente comparados.

	F1-score	Accuracy
Línea base	0.50	0.51
Reglas	0.62	0.62
Aprendizaje automático	0.61	0.61
Híbrido	**0.64**	**0.64**

Cuadro 7.17: Resumen de resultados

Se logra por lo tanto con los clasificadores creados una mejora de 13 o 14 porciento (en accuracy y F1-score promedio respectivamente) de un 22 porciento posible, ya que 72 porciento eran los valores de la línea tope.

Además, se puede ver como la creación del clasificador híbrido logra mejorar los resultados obtenidos únicamente con reglas y únicamente con aprendizaje automático, capturando parte de las virtudes de ambos métodos.

7.3. Análisis de los errores

El estudio de los errores cometidos es importante tanto para el mejor entendimiento del comportamiento del clasificador como para conocer sus posibles puntos de mejora.

Además del continuo análisis de errores en el conjunto de entrenamiento realizado a lo largo de todo el proyecto, se realiza un análisis de los errores en el conjunto de prueba, que se describe en esta sección.

7.3.1. Clasificación de los errores

Se analizó caso a caso los errores cometidos por el clasificador, de forma de cuantificar la influencia de cada una de las dificultades del problema en la cantidad total de errores. El análisis consistió en clasificar manualmente cada uno de los errores en las categorías de problemas descritos en la sección 3.2.

De las 335 opiniones erróneamente clasificadas (35.7 % del total), 50 (5.3 %) fueron en realidad clasificadas con una polaridad de las dos posibles, como fue explicado en la sección 7.2.7 y no pueden clasificarse como errores. Eso nos deja 285 fallos (30.4 % del total de opiniones) de los cuales:

- **42 (14.7 %)** se debieron a problemas de falta de contexto textual, temporal, político o socio-cultural.
- **36 (12.6 %)** fueron por la presencia de múltiples polaridades en la opinión, y donde el clasificador tomó una decisión distinta a la que fue tomada al momento de anotar sobre cuál considerar como la principal.
- **25 (8.8 %)** a la presencia de múltiples opiniones con distinta polaridad dentro del texto y nuevamente una decisión distinta que los anotadores al seleccionar la principal
- **3 (1.1 %)** fueron causados por presencia de ironía no detectada por el clasificador.

Las 178 restantes (62.5 %) se debieron a errores del clasificador, sin que pudiera ser detectada la causa de los mismos. Esas 178 opiniones representan el 19.2 % del total del conjunto de test.

7.3.2. Expresiones no consideradas

Varias opiniones fueron incorrectamente clasificadas debido a que ciertas expresiones que contienen polaridad no fueron consideradas por el algoritmo. Esto se debe a que no están presentes en el conjunto de entrenamiento ni en las listas de palabras.

Algunas de ellas fueron:

- *hecho trizas*
- *saludable*
- *tener atado*
- *La Cenicienta de*
- *quedar por el camino*
- *Museo muerto*

- *subirse al estribo*
- *maquiavélico*
- *al cuete*
- *cuetazo*

Se podría agregar las expresiones en las listas o incluir oraciones que las contengan en el entrenamiento para que sean tenidas en cuenta y clasificar correctamente estas opiniones. Sin embargo, eso sería sobreajustar por lo que no se hace para no mejorar artificialmente los resultados.

Capítulo 8

Integración con BuscOpiniones

Buscopiniones [25, 28] es una herramienta desarrollada con el objetivo de poder recolectar, buscar y visualizar opiniones de una fuente elegida. Las opiniones son obtenidas de una base de datos generada a partir de artículos de prensa publicados en portales de noticias uruguayos.

Su funcionamiento básico como herramienta web ofrece la posibilidad de introducir un nombre (*Mujica*, *Tabaré Vázquez*, etc) o identificador de una posible fuente (*presidente*, *intendente*, *director técnico*, etc) y el tema del que se quiere recuperar opiniones (*marihuana*, *transporte*, etc). Luego, mediante el envío de esos datos al servidor, se obtiene y visualiza en una línea de tiempo las opiniones recolectadas por la plataforma de esa fuente y sobre ese tema.

El sistema central consta de tres módulos principales: el que recolecta los artículos de la web y los indexa, el que permite realizar las búsquedas y desplegarlas en el navegador, y el que extrae las opiniones de cada artículo.

La recolección se hace a través de un scraper de noticias de las páginas web de los diarios El País, La República y El Observador. Las noticias luego son procesadas e indexadas en *Solr* [1].

Para permitir la búsqueda interactiva de opiniones, el sistema cuenta con un servidor web donde se encuentra instalada la aplicación, al cual se puede acceder desde un navegador. En la interfaz desplegada tiene dos campos que permiten realizar la búsqueda, uno para la fuente y otro para el tema de la opinión, además de campos de búsqueda avanzada. Una vez ingresados los valores deseados, la web despliega en una línea de tiempo las opiniones extraídas por el módulo de extracción de opiniones, de forma cronológica.

El módulo de extracción de opiniones de los artículos es la columna vertebral del sistema. El trabajo de reconocer qué partes de los textos son opiniones y la obtención de la fuente que emitió la opinión, fueron la motivación inicial del desarrollo de la plataforma, y donde se encuentra la aplicación del Procesamiento de Lenguaje Natural.

8.1. Módulo de clasificación de sentimiento

Para agregar la clasificación de sentimiento al sistema, se crea un módulo de clasificación que se comunica directamente con el módulo que realiza las búsquedas. De esta manera, cada vez que se realiza una búsqueda y se retorna un conjunto de opiniones, para cada una de las opiniones se retorna su polaridad.

Nuevamente existen dos alternativas principales a la hora de retornar la polaridad de una opinión. La primera es enviar cada opinión al clasificador de sentimiento desarrollado y calcular su polaridad cada vez que se realiza una búsqueda. La segunda, es que el valor de sentimiento ya haya sido calculado alguna vez e indexado, de forma que cuando se retorna el valor almacenado el costo de procesamiento es mínimo.

Este proyecto utiliza la primera alternativa, aunque se aprovecha cada búsqueda para, una vez calculada la polaridad, indexarla para no hacerlo las veces siguientes. De esta manera las opiniones más populares no tendrán la sobrecarga generada por el tiempo de clasificación, ya este proceso se realiza sólo la primera vez que son recuperadas.

Una solución más con mejor desempeño para el usuario del sistema sería realizar la clasificación al momento de recuperar las noticias e indexarlas. Además, hacer un pre-procesamiento de todas las opiniones ya existentes en el sistema de forma de evitar la sobrecarga temporal en las búsquedas. Sin embargo, queda como trabajo a futuro ya que no se consideró prioritario en el proyecto y de todas formas el programa no pierde funcionalidad.

BuscOpiniones guarda sus datos utilizando el motor de búsqueda *Solr*. Con los objetivos de dedicar los menores recursos posibles en rehacer un trabajo que ya estaba bien resuelto y no perder los datos ya recolectados, que pueden haber sido incluso eliminados de la web, se consideró como mejor opción mantener este esquema. Sin embargo, el esquema creado en *Solr* para archivar los datos no incluía un lugar para la información del sentimiento, por lo tanto tuvo que ser agregado.

8.2. Almacenamiento de la polaridad

Una opción podría haber sido mantener intacto el esquema de *Solr* ya existente y analizar cada una de las opiniones a demanda. Esta opción fue tomada en cuenta y descartada por dos razones: la pérdida de velocidad y la pérdida de flexibilidad. En primer lugar, cada vez que se recuperara una opinión habría que procesar el análisis para recién luego devolver el resultado; teniendo en cuenta que algunas búsquedas devuelven decenas de resultados, el tiempo podría haber sido considerablemente peor que al haberlas analizado previamente. Además, en caso de buscar, por ejemplo, opiniones únicamente positivas sobre un tema, se debería haber

recuperado todas las opiniones y luego filtrado las negativas y neutrales, generando una triple recarga de tiempo: recuperar más opiniones de las necesarias, analizar las opiniones sobrantes y quitar las opiniones que no servían.

Se consideran por estas razones dos opciones disponibles: modificar el esquema de *Solr* o agregarlo como un campo en un archivo externo [4].

La primera de las alternativas da la posibilidad de mantener la simplicidad del esquema Solr, simplemente agregando un campo. Además, Solr es flexible en permitir los cambios y mantener vigentes los documentos ya ingresados pese a no contar con el nuevo campo. Por otro lado, para sumar el campo sentimiento a cada uno de los documentos ya ingresados se debería haber actualizado cada uno de ellos.

La otra opción es agregar en el esquema una referencia a un archivo externo que, a cada documento referenciado por su id, le asigne un valor para el campo sentimiento. Esta alternativa del archivo externo tiene el problema de la performance: el desempeño es peor ya que *Solr* debe abrir el archivo externo y buscar el valor correspondiente cada vez que retorna un documento.

Sin embargo, se considera que el pequeño deterioro en el rendimiento no logra equiparar las virtudes representadas por la opción del archivo externo. La principal ventaja tomada en cuenta es la posibilidad de acceder a cambiar o agregar resultados de sentimiento al sistema sin siquiera interactuar mínimamente con *Solr* o *BuscOpiniones*, sino simplemente accediendo a un archivo. En el planeamiento del proyecto, se tuvo especial atención en la idea de aislar el proceso de análisis de las opiniones con el resto, por lo que es un paso muy importante ya que permite prescindir incluso de la comunicación entre los dos módulos. Es por esto que se decide llevar a cabo la opción de agregar en el esquema una referencia a un archivo externo.

8.3. Interfaz

Para diferenciar las opiniones que se muestran positivas, negativas y neutras en la web, se le asigna un color a cada opinión: verde, rojo y gris respectivamente. De esta manera, el usuario logra interpretar la polaridad de la opinión de manera intuitiva y fácil como se ven la figura 8.1.

A su vez, se considera de utilidad agregar la funcionalidad de búsqueda avanzada por polaridad. De esta manera se pueden buscar opiniones únicamente positivas, negativas o neutras sobre un tema en particular. En la figura 8.2 se puede observar gráficamente el campo agregado.

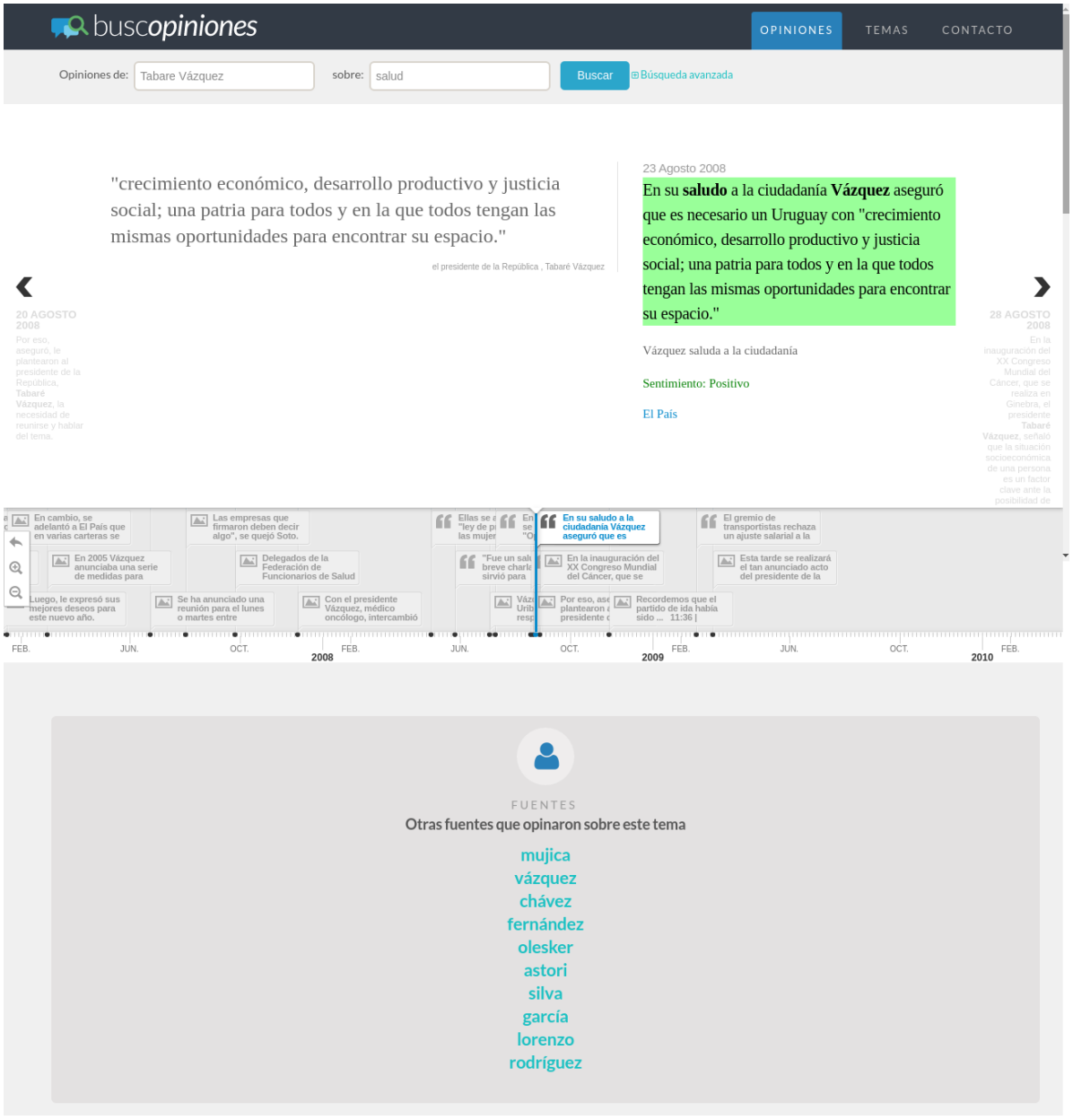

FIGURA 8.1: Polaridad de una opinión mostrada en *BuscOpiniones*

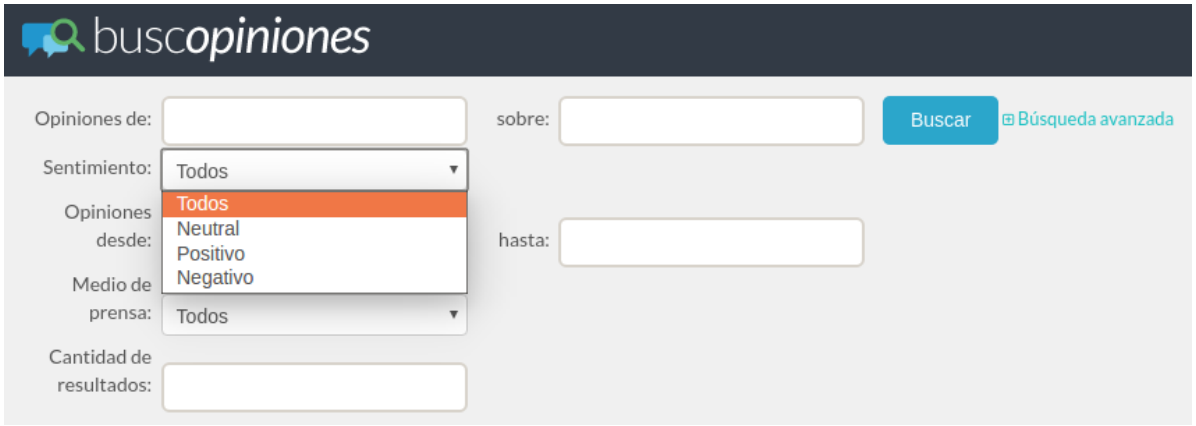

FIGURA 8.2: Campo de búsqueda de sentimiento en *BuscOpiniones*

Capítulo 9

Conclusiones y trabajo futuro

Este capítulo presenta las conclusiones del trabajo realizado en el marco del proyecto y describe las posibles mejoras a realizar en el futuro.

9.1. Conclusiones

En este proyecto se trabajó en la construcción de un algoritmo de análisis de sentimiento capaz de clasificar la polaridad de opiniones como positivas, neutrales o negativas.

Con ese objetivo, se estudió el marco teórico del problema en el contexto del Procesamiento de Lenguaje Natural. En particular, se analizó el estado del arte de las soluciones del problema, tanto en su rama de procesamiento mediante reglas como de aprendizaje automático.

Además, por ser el análisis de sentimiento un problema fuertemente relacionado a los textos específicos sobre los que se aplica, junto con su contexto, se estudiaron las particularidades de los textos de prensa de Uruguay. Incluso, debido a que no existía ningún corpus que se ajustara a este proyecto, se generó un corpus anotado tanto de prueba como para entrenamiento.

Utilizando el conocimiento adquirido, se crearon tres clasificadores: uno con el enfoque de reglas, otro exclusivamente basado en aprendizaje automático y uno con un enfoque híbrido que combina los anteriores. Este último enfoque fue el que generó los mejores resultados ya que se logró tomar lo mejor de cada uno de los dos métodos alimentando al algoritmo de aprendizaje automático con la información obtenida mediante las reglas.

Los resultados superaron ampliamente los obtenidos por la línea base (51 % de accuracy, 50 % de F1-score) con un 64 % tanto de accuracy total como de F1-score promedio en las tres clases. Estos resultados son además cercanos a la línea tope marcada por la concordancia entre anotadores del corpus de prueba (72 % de accuracy, 72 % de F1-score). Principalmente, se está

muy cerca de la línea tope si, tomando en cuenta que la clasificación manual de ciertas opiniones es dudosa entre dos categorías (un 13 % del total), se toma como correcta la clasificación en cualquiera de esas dos opciones. Teniendo esa tolerancia en las opiniones dudosas se llega a 69 % de accuracy y 70 % de F1-score promedio.

Junto con el desarrollo de los clasificadores, se obtuvieron y generaron varios recursos valiosos tanto para el proyecto como para el grupo de PLN de la Facultad de Ingeniería para futuros trabajos. Uno de ellos es el ya mencionado corpus anotado de opiniones de prensa uruguaya. Además, se obtienen listas de palabras con su polaridad semántica y se las adapta a la variante del español utilizado en Uruguay. También se consigue un parser morfo-sintáctico del idioma español llamado MateParser que obtiene mejores resultados que los utilizados en otros proyectos [20]. En los dos últimos casos, se obtuvieron mediante la interacción con los autores de cada uno de ellos, generando un vínculo tanto con los integrantes del proyecto como con el grupo de PLN de la facultad.

Finalmente, se integró el algoritmo creado al sistema BuscOpiniones para que se pueda visualizar el sentimiento de las opiniones además de realizar búsquedas, todo de forma simple e intuitiva.

Por lo tanto se cumplen con todos los objetivos propuestos (sección 1.2), generando recursos valiosos que podrán ser reutilizados y desarrollando un algoritmo de clasificación de sentimiento integrado a un sistema de búsqueda de opiniones que estará a disposición del público para ser utilizado por quienes lo deseen.

9.2. Trabajo Futuro

Existen muchas mejoras posibles en todas las áreas del proyecto, que se describen en esta sección.

Principalmente, vale notar que no es posible ser exhaustivo en cuanto a las palabras con polaridad semántica en un lenguaje natural ni mucho menos si se toma en cuenta cada uno de los contextos donde pueden aparecer y las variaciones que éstos generan. Tampoco se puede completar las reglas que dominan el idioma, entre otras razones porque el idioma cambia. Por lo tanto, siempre se pueden perfeccionar y aumentar los recursos léxicos.

De la misma forma, cuantos más ejemplos de entrenamiento tenga un algoritmo de aprendizaje automático, asumiendo que son buenos ejemplos correctamente clasificados, mejor será su aprendizaje. Por esta razón siempre es posible agregar ejemplos al corpus para mejorar los resultados.

Sin embargo, existen algunos cambios que van más allá de agregar recursos, como la mejora en la detección del tema. Como fue explicado anteriormente, BuscOpiniones permite búsquedas de opiniones según el tópico, pero lo hace realizando búsquedas de palabras. Crear un algoritmo que clasifique el tema de opinión de forma más inteligente sería muy beneficioso para utilizar esa información como entrada del sistema de análisis de sentimiento.

También se pueden generar clasificadores de ironía y principalmente de opiniones *irrealis* (como se definien en la sección 2.2) o *insustanciales* o *irrealis* (como se definen en el apéndice A). En ese caso podría lograrse que no sean tenidas en cuenta al ser clasificados los textos que las incluyen. Esto mejoraría en buena parte los resultados reduciendo varios de los errores más comunes.

Además, como se explicó en los criterios de anotación en la sección 3.2, muchos de los textos con los que se trabajó en este proyecto no son en realidad opinados. La detección de opiniones en los textos no es una tarea sencilla y constituye todo una rama del PLN. Sin embargo, una clasificación más acertada en este aspecto que la que realiza BuscOpiniones podría ayudar, principalmente en los resultados de la categoría neutral que a lo largo de todo este proyecto fue en realidad *neutral o no opinado*.

Con el objetivo también de mejorar el clasificador, queda como trabajo futuro la implementación de un método basado en redes neuronales o aprendizaje profundo. Este tipo de algoritmos, muy utilizados recientemente en este tipo de problemas, han dado buenos resultados y por lo tanto podrían ser de utilidad. A pesar de esto, usualmente requieren una cantidad de datos mucho más grande que las que requieren MNB o SVM, por lo que para utilizarlos se debería aumentar el corpus de entrenamiento.

Finalmente, existen posibles mejoras en la integración con BuscOpiniones. Seguramente lo principal sea disminuir el tiempo de ejecución del algoritmo debido a que, principalmente por el tiempo que consume el parser, cada clasificación insume un tiempo del orden de los segundos cuando se realiza por primera vez. Este efecto se logró mitigar con el caché, pero lo ideal sería hacerlo de forma de que insumiera menos tiempo o realizarla al momento de recuperar la opinión y no cuando es buscada.

Bibliografía

[1] *Apache Solr*. http://lucene.apache.org/solr/.

[2] *Feature extraction, Scikit-Learn*. http://scikit-learn.org/stable/modules/feature_extraction.html.

[3] *Google DeepMind, AlphaGo*. https://deepmind.com/alpha-go.

[4] *Solr: Working with External Files and Processes*. https://cwiki.apache.org/confluence/display/solr/Working+with+External+Files+and+Processes.

[5] Ahmad, Khurshid: *Affective Computing and Sentiment Analysis: Emotion, Metaphor and Terminology - Volume 45*. Springer Publishing Company, Incorporated, 2013.

[6] Björkelund, Anders, Bernd Bohnet, Love Hafdell y Pierre Nugues: *A High-performance Syntactic and Semantic Dependency Parser*. En *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, COLING '10, páginas 33–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1944284.1944293.

[7] Cambria, Erik, Bjorn Schuller, Yunqing Xia y Catherine Havasi: *New Avenues in Opinion Mining and Sentiment Analysis*. IEEE Intelligent Systems, 28(2):15–21, Marzo 2013, ISSN 1541-1672. http://dx.doi.org/10.1109/MIS.2013.30.

[8] Choi, Yejin y Claire Cardie: *Learning with Compositional Semantics As Structural Inference for Subsentential Sentiment Analysis*. En *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, páginas 793–801. Association for Computational Linguistics, 2008. http://dl.acm.org/citation.cfm?id=1613715.1613816.

[9] Das, Sanjiv R., Mike Y. Chen, To Vikas Agarwal, Chris Brooks, Yuk shee Chan, David Gibson, David Leinweber, Asis Martinez-jerez, Priya Raghubir, Sridhar Rajagopalan, Ajit Ranade, Mark Rubinstein y Peter Tufano: *Yahoo! for amazon: Sentiment extraction from small talk on the web*. En *8th Asia Pacific Finance Association Annual Conference*, 2001.

[10] Fleiss, J.L. *y cols.*: *Measuring nominal scale agreement among many raters.* 76(5):378–382, 1971.

[11] Gwet, Kilem: *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring The Extent of Agreement Among Raters.* Advanced Analytics, LLC., 2014.

[12] Hatzivassiloglou, Vasileios y Kathleen R. McKeown: *Predicting the Semantic Orientation of Adjectives.* páginas 174–181, 1997.

[13] Johnson-Laird, P. N. y Keith Oatley: *The Language of Emotions: An Analysis of a Semantic Field.* Cognition and Emotion, 3(2):81–123, 1989.

[14] Jurafsky, Dan y James H. Martin: *Speech & language processing.* Pearson Education India, 2000.

[15] Kaufer, David S: *The power of words : unveiling the speaker and writer's hidden craft.* Mahwah, N.J. : Lawrence Erlbaum, 2004.

[16] Liu, Bing: *Sentiment Analysis and Subjectivity.* En Indurkhya, Nitin y Fred J. Damerau (editores): *Handbook of Natural Language Processing, Second Edition.* CRC Press, Taylor and Francis Group, 2010.

[17] Liu, Bing: *Sentiment Analysis and Subjectivity.* En Indurkhya, Nitin y Fred J. Damerau (editores): *Handbook of Natural Language Processing, Second Edition*, páginas 13–14. CRC Press, Taylor and Francis Group, 2010.

[18] Mitchell, Thomas M.: *Machine Learning.* McGraw-Hill, Inc., 1ª edición, 1997.

[19] Moilanen, Karo y Stephen Pulman: *Sentiment Composition.* En *Proceedings of Recent Advances in Natural Language Processing*, páginas 378–382, 2007. `http://users.ox.ac.uk/~wolf2244/sentCompRANLP07Final.pdf`.

[20] Padró, Muntsa, Miguel Ballesteros, Héctor Martínez y Bernd Bohnet: *Finding dependency parsing limits over a large Spanish corpus.* En *Proceedings of 6th International Joint Conference on Natural Language Processing*, Nagoya, Japan, October 2013. `http://www.taln.upf.edu/system/files/biblio_files/ijcnlp_final_padro_et_al_2013.pdf`.

[21] Pang, Bo y Lillian Lee: *Opinion Mining and Sentiment Analysis.* Found. Trends Inf. Retr., 2(1-2):1–135, Enero 2008, ISSN 1554-0669. `http://dx.doi.org/10.1561/1500000011`.

[22] Pang, Bo, Lillian Lee y Shivakumar Vaithyanathan: *Thumbs Up? Sentiment Classification Using Machine Learning Techniques.* En *Proceedings of EMNLP*, páginas 79–86, 2002.

[23] Polanyi, Livia y Annie Zaenen: *Contextual Valence Shifters*, páginas 1–10. Springer Netherlands, 2006. `http://dx.doi.org/10.1007/1-4020-4102-0_1`.

[24] Quirk, Randolph, Sidney Greenbaum, Geoffrey Leech y Jan Svartvik: *A Comprehensive Grammar of the English Language*. Longman, London, 1985.

[25] Rosá, Aiala: *Identificación de opiniones de diferentes fuentes en textos en español*. mastersphd, Proyecto de Apoyo a las Ciencias Básicas - Universidad de la República, Montevideo, Uruguay - École Doctorale Connaissance, Langage, Modélisation Université Paris Ouest, Nanterre, La Défense, 2011. `http://www.fing.edu.uy/inco/grupos/pln/publicaciones/rosa2011.pdf`.

[26] Rosá, Aiala, Dina Wonsever y Jean Luc Minel: *Combining Rules and CRF Learning for Opinion Source Identification in Spanish Texts*, páginas 452–461. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ISBN 978-3-642-34654-5. `http://dx.doi.org/10.1007/978-3-642-34654-5_46`.

[27] Shalev-Shwartz, Shai y Shai Ben-David: *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[28] Stecanella, Rodrigo, Jairo Bonanata, Dina Wonsever y Aiala Rosá: *Opinion Search in Spanish Written Press*, páginas 307–318. Springer International Publishing, Cham, 2014, ISBN 978-3-319-12027-0. `http://dx.doi.org/10.1007/978-3-319-12027-0_25`.

[29] Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll y Manfred Stede: *Lexicon-based Methods for Sentiment Analysis*. Comput. Linguist., 37(2):267–307, Junio 2011, ISSN 0891-2017. `http://dx.doi.org/10.1162/COLI_a_00049`.

[30] Turney, Peter D. y Michael L. Littman: *Measuring Praise and Criticism: Inference of Semantic Orientation from Association*. 21(4):315–346, Octubre 2003. `http://doi.acm.org/10.1145/944012.944013`.

[31] White, Michele: *Representations or people?* Ethics and Information Technology, 4(3):249–266, 2002, ISSN 1572-8439. `http://dx.doi.org/10.1023/A:1021376727933`.

[32] Wiebe, Janyce, Theresa Wilson y Matthew Bell: *Identifying Collocations for Recognizing Opinions*. En *In Proc. ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, páginas 24–31, 2001.

[33] Wiebe, Janyce, Theresa Wilson y Claire Cardie: *Annotating Expressions of Opinions and Emotions in Language*. Language Resources and Evaluation, 39(2):165–210, 2005.

Glosario

- **Categoría gramatical**: Clasificación de una palabra según su función dentro de una oración (verbo, sustantivo, adjetivo, etc.).
- **Corpus**: Conjunto estructurado de textos que generalmente representan ejemplos reales de uso de una lengua dentro de un contexto.
- **Etiquetado morfológico**: Proceso de asignar (o etiquetar) a cada una de las palabras de un texto su categoría gramatical.
- **Ejemplo**: Instancia perteneciente a un conjunto de datos que se quiere estudiar.
- **Feature**: Propiedad mensurable de una instancia que se observa.
- **Grafo**: Representación simbólica de los elementos constituidos de un sistema o conjunto, mediante esquemas gráficos.
- **Grafo acíclico**: Es un grafo dirigido G que no tiene ciclos, o sea, para cada vértice v en G, no hay un camino directo que empiece y termine en v.
- **Grafo conexo**: Es un grafo G que cumple que si para cualquier par de vértices u y v en G, existe al menos una trayectoria (una sucesión de vértices adyacentes que no repita vértices) de u a v.
- **Grafo dirigido**: Es un tipo de grafo en el cual las aristas tienen un sentido definido.
- **Línea Base**: Algoritmo simple pero razonable utilizado para establecer el rendimiento mínimo esperado en un conjunto de datos.
- **Lexicón**: Serie ordenada de palabras de una lengua, una persona, una región, una materia o una época determinadas.
- **Matriz de confusión**: Tabla para visualizar el resultado de una clasificación, en donde cada columna representa la cantidad de predicciones de cada clase y cada fila indica la cantidad real de cada una. Permite ver por ejemplo bajo qué clases fueron clasificadas las instancias de una clase, así como a qué clases pertenecían realmente las instancias predecidas como de una categoría dada.

- **MNB**: Multinomial Naïve Bayes.
- **Parser**: Es una de las partes de un compilador que transforma su entrada en un árbol de derivación.
- **SVM**: Support Vector Machine.
- **N-grama**: N-grama Es un modelo de lenguaje que determina la probabilidad de ocurrencia de una palabra en base a las n - 1 palabras anteriores.
- **Token**: Símbolo utilizado como unidad mínima de trabajo en el análisis de cierto texto. En este trabajo un token equivale a una palabra en una opinión.
- **Tokenización**: Proceso de separar un texto en tokens.

Apéndice A

Detalle del estudio del corpus

Como parte de la exploración inicial y análisis del problema, se realizó una anotación manual de oraciones. Esto consiste en analizar un conjunto de oraciones que expresan un estado privado, de forma de identificar el sentimiento de las mismas. Además, se marcaron los principales elementos que hacen al anotador percibir la subjetividad de la oración, así como la posición de su autor sobre el tema del que habla.

A.1. Motivación

El principal objetivo de esta etapa fue profundizar en los obstáculos y las claves del problema a resolver, junto con el estudio del marco teórico del trabajo, como fue descrito en el capítulo 2.

No fue una meta de esta etapa la creación de un corpus anotado en español, sino que se consideró un paso previo de investigación y acuerdo de criterios. Sin embargo, cabe mencionar que la notación propuesta y utilizada para esta tarea, así como el método de trabajo, podría servir como base para la implementación en trabajos futuros de una anotación que necesitara de la granularidad a la que se llegó.

A.2. Esquema de anotación

El esquema de anotación utilizado fue basado en el propuesto por Wiebe et al. [33]. La anotación refinada, más allá del nivel de oración y con registro además del resultado final de los pasos y señales que llevan a él, hacen que se ajuste perfectamente a lo buscado en esta etapa.

El esquema se basa en marcos que anotan las expresiones manifestadas por una fuente (una persona, organización, etc). Se marcan tanto las oraciones objetivas como las que expresan un *estado privado*: «un término general que cubre opiniones, creencias, pensamientos, sentimientos, emociones, objetivos, evaluaciones y juicios», como es descrito en el artículo.

Los marcos objetivos señalan las expresiones que son atribuidas a una fuente pero no contienen subjetividad. El objetivo de marcar este tipo de expresiones es el de distinguirlas de las subjetivas, ya sea con fines estadísticos (por ejemplo, para estimar la probabilidad a priori de que una oración sea opinada en un texto) o de búsqueda de similitudes o diferencias cualitativas que ayuden en la posterior decisión. Este tipo de discernimiento escapaba el objetivo del trabajo, ya que se tiene como entrada el sistema BuscOpiniones que ya lo realiza. Igualmente, se consideró que marcar los sintagmas objetivos podría tener valor y se hizo de todas formas.

Los sintagmas subjetivos, los que expresan un estado privado, se identifican con un marco subjetivo, que puede contener hasta siete propiedades. Lo primero que se marca es el *ancla*: el texto que se está anotando. Luego el objeto de la subjetividad, sobre qué se está opinando, y la fuente, quién lo está haciendo. Estas tres marcas tampoco pertenecen al alcance del proyecto ya que BuscOpiniones incluye la recuperación de opiniones en el texto, obteniendo la fuente y el objeto de las mismas. De todas formas se deben encontrar ya que sin esta información muchas veces no es posible notar la polaridad, ya que se pierde el contexto en el que la opinión fue emitida.

La anotación de la fuente tiene la particularidad de poder ser anidada. Esto permite marcar cuando una fuente cita a otra o traslada un estado privado ajeno. Por ejemplo, si el escritor dice «Mujica siempre estuvo a favor de la despenalización de la marihuana», la fuente que opina sería <Escritor, Mujica> marcando que no necesariamente esa es la opinión de Mujica, sino lo que el escritor considera la opinión de Mujica. De la misma forma, en la oración «el presidente Venezolano Hugo Chávez le respondió a Zapatero “me refiero a José Gervasio Artigas cuando dijo: ‘Con la verdad ni ofendo ni temo’”» la fuente que se expresa positivamente sobre la verdad es <Escritor, Chávez, Artigas>. En este caso, se ve claramente la necesidad de la anidación, ya que en realidad *Artigas* no se refería a la verdad sino a la libertad, por lo que en caso de anotar la opinión con *Artigas* como fuente se hubiera cometido un error.

Las otras cuatro propiedades eran las que más directamente interesaban en este proyecto. El «tipo de actitud» se refiere a la polaridad. El artículo sugiere como valores “positivo”, “negativo”, “ninguno” y “otro”, aunque se decide sustituirlos por “positivo”, “negativo” y “neutral” de forma de adecuarlo este proyecto. La polaridad se complementa con la intensidad, otra de las propiedades del marco, que puede tomar los valores “bajo”, “medio”, “alto” o “muy alto”. Este valor sirve para marcar el grado de fuerza de una opinión positiva o negativa, es decir, cuán cerca está de la neutralidad o de los extremos. Esta información, además de tener

un valor en sí misma, puede ayudar a encontrar casos borde como en los cuales se duda entre la neutralidad y una polarización de intensidad baja.

Además de la polaridad de la oración en sí, se cuenta con una propiedad opcional llamada «polaridad de la expresión» que permite marcar la subjetividad de la expresión del acto del habla. Por ejemplo, si la opinión se cita con un "dijo" es neutral pero si se hace con "lamentó" es negativa.

Finalmente, está la propiedad de *insustancial*, que sirve para marcar que una opinión es condicional o una suposición y no debería ser tomada en cuenta como real. Por ejemplo, en la expresión «Según Lacalle Pou, con el mismo *ímpetu* que el gobierno combate el tabaco, mantiene *tolerancia* hacia la marihuana.» la opinión con fuente <Escritor, Lacalle Pou, gobierno> sobre la marihuana y que expresa que se *mantiene tolerancia* es insustancial, ya que no refleja lo que opina el gobierno sobre la marihuana, sino lo que Lacalle Pou opina que el gobierno opina.

Además, el esquema cuenta con los elementos de expresión subjetiva que sirven para marcar expresiones que no son una opinión en sí, sino que definen, matizan o reafirman la subjetividad de una opinión. Tienen menos propiedades que los marcos subjetivos, pero al igual que ellos poseen ancla del texto, fuente, tipo de actitud e intensidad.

De esta forma, se puede descubrir y documentar qué es lo que hace que se pueda identificar la actitud y la intensidad de la opinión.

A.3. Opiniones a anotar

Las oraciones fueron tomadas de BuscOpiniones y se determinó conveniente que las mismas fueran principalmente políticas, debido a que es el tema con más ejemplos en el corpus. Sin embargo, no en su totalidad de política con el objetivo de evitar el sesgo que esto podría haber generado. Por lo tanto, se tomaron también opiniones deportivas.

Por otra parte, vale notar que, como se explicó anteriormente, los problemas resueltos en el ámbito del procesamiento de lenguaje natural no logran alcanzar precisiones del cien por ciento, sino que indefectiblemente se comenten errores. Por esta razón, los datos de entrada del sistema a construir contienen errores tales como oraciones objetivas o marcado erróneo de emisor y/o sujeto de la opinión. Por lo tanto, se decidió tomar todas las oraciones que aparecieran en cada una de las búsquedas y no únicamente las más completas y bien formadas, con el objetivo de saltear la menor cantidad de problemas cometidos por BuscOpiniones. Se consideró positivo evaluar los problemas de este tipo con los que habría que enfrentarse en etapas posteriores, y cómo estos podrían influir.

Se seleccionaron los temas de forma que el sistema encontrara varias opiniones y que fueran representativos del conjunto total. Los temas elegidos fueron: *«**Ana Olivera** sobre **transporte**»*, *«**Bordaberry** sobre **educación**»*, *«**Vázquez** sobre **el aborto**»* y *«**Tabárez** sobre **Suárez**»*.

A.4. Primera etapa de anotación

Una vez definidas las opiniones, cada uno de los tres integrantes del proyecto anotó por separado las diez primeras opiniones tomando como base el sistema de anotación acordado. Esta fase inicial se hizo sin comunicación entre integrantes para no influenciar la forma de pensar de cada uno y finalmente se realizó una comparación de resultados para unificar criterios.

En líneas generales existió un consenso en las opiniones anotadas, aunque hubo discrepancias en algunas de ellas, principalmente surgidas por algunas ambigüedades y problemas encontrados.

Por lo tanto, se fijaron ciertos criterios para uniformizar la anotación que se respetaron durante todo el proyecto y se describirán en detalle en la sección 3.2.

A.5. Cambios en el esquema de anotación

Luego de la primera etapa de anotación, se notó que el esquema original propuesto por [33] no permite expresar algunas de las propiedades que se buscaba marcar en las opiniones.

Por esa razón, se decidió modificar el esquema de forma de poder mejorar la expresividad. Fundamentalmente, las diferencias de este nuevo esquema son una estructura arborescente, que permite reflejar las relaciones asimétricas entre los componentes de las opiniones, y la posibilidad de marcar los conectores y las implicancias que estos tienen en la polaridad.

A.5.1. Estructura arborescente

La notación propuesta por el artículo tiene una estructura lineal, sin que existan relaciones asimétricas entre ninguno de los marcos. Este hecho genera por ejemplo que en presencia de más de un marco subjetivo, no se pueda identificar con facilidad con cuál de ellos se relaciona

cada una de las expresiones subjetivas.

> *Poco antes, Mujica afirmó que su proyecto de legalizar la compraventa de marihuana en el país tiene por objetivo combatir el narcotráfico, al que definió como «el peor flagelo de América Latina»* (A.1)
>
> Mujica

Para la opinión A.1, con el nuevo esquema arborescente, la anotación es la siguiente: **Objective speech event:** *Text anchor:* the entire sentence *Source:* <writer> *Implicit:* true

Direct subjective: *Text anchor:* afirmó *Source:* <writer, Mujica> *Intensity:* extreme *Expression intensity:* neutral *Target:* narcotráfico *Attitude type:* negative

- **Expressive subjective element:** *Text anchor:* combatir *Source:* <writer, Mujica> *Intensity:* High *Attitude type:* negative
- **Expressive subjective element:** *Text anchor:* peor *Source:* Mujica *Intensity:* high *Attitude type:* negative
- **Expressive subjective element:** *Text anchor:* flagelo *Source:* Mujica *Intensity:* extreme *Attitude type:* negative

Donde se ve que los tres elementos de expresión subjetiva corresponden a *Mujica* hablando negativamente sobre *el narcotráfico*.

A.5.2. Sesgo de objetividades que generan subjetividad

Como fue explicado en la sección 3.2.5, se notó que en ciertas ocasiones la selección tendenciosa de propiedades objetivas genera una subjetividad. Esto no se puede anotar de forma correcta con el anterior esquema por lo que se considera necesario crear los *elementos de expresión objetiva*, que pueden formar parte de los hijos de las expresiones subjetivas incluso en exclusividad.

> *Para Lacalle Pou, las medidas que está empleando el gobierno en relación a la economía van a generar más déficit, más recesión y van a encarecer el costo de vida.* (A.2)
>
> Lacalle Pou

En la opinión A.2 la anotación con este cambio es la siguiente:

Objective speech event *Text anchor:* the entire sentence *Source:* <writer> *Implicit:* true **Direct subjective** *Text anchor:* Para *Source:* <writer, Lacalle Pou> *Intensity:* high *Expression intensity:* neutral *Target:* medidas que está empleando el gobierno en relación a la economía *Attitude type:* negative

- **Expressive objective element** *Text anchor:* más déficit *Source:* <writer, Lacalle Pou> *Intensity:* High *Attitude type:* negative
- **Expressive objective element** *Text anchor:* más recesión *Source:* <writer, Lacalle Pou> *Intensity:* High *Attitude type:* negative
- **Expressive objective element** *Text anchor:* encarecer el costo de vida *Source:* <writer, Lacalle Pou> *Intensity:* High *Attitude type:* negative

Si bien en este caso puede ser discutible la objetividad de las tres afirmaciones sobre las consecuencias de las medidas económicas, es el hecho de que todas sean negativas lo que genera la subjetividad. Si en lugar de un político se tratara de un economista que en un análisis dijera «las medidas que está empleando el gobierno en relación a la economía van a disminuir el déficit, generar más recesión y van a abaratar el costo de vida», la mixtura entre los efectos positivos y negativos haría catalogar la oración como objetiva. Por lo tanto, se concluye que cada una de las consecuencias pueden ser consideradas objetivas por separado, no lo son en su conjunto.

A.5.3. Conectores subjetivos

Algunos conectores, tales como los de contraste y los de consecuencia agregan subjetividad a las oraciones, como se detalla en la sección 2.2, y eso no era posible de marcar con el esquema de anotación original. En particular, se notó que muchas de las opiniones negativas, son precedidas por una parte positiva y un conector de contraste como el pero, de forma de suavizar la crítica. Seguramente esto se deba al tipo de opiniones, de carácter en su mayoría político.

Por esa razón, se crearon los conectores subjetivos.

El presidente José Mujica dice que legalizará la marihuana, pero después envía al Parlamento un proyecto de ley donde no se detalla nada y todos dan por hecho que nunca se concretará. (A.3)

Lacalle Pou

Un ejemplo de su utilización se puede ver en la opinión A.3, que tendrá con este nuevo esquema la siguiente anotación:

Direct subjective *Text anchor:* the entire sentence *Source:* <writer> *Intensity:* high *Expression intensity:* neutral *Target:* Mujica *Attitude type:* negative **Subjective connector** *Text anchor:* pero *Source:* <writer> *Type:* Adversativas

- **First:**
  - **Objective speech event** *Text anchor:* El presidente José Mujica dice que legalizará la marihuana *Source:* <writer> *Implicit:* true
- **Second:**
  - **Direct subjective** *Text anchor:* después envía al Parlamento un proyecto de ley donde no se detalla nada y todos dan por hecho que nunca se concretará. *Source:* <writer> *Intensity:* high *Expression intensity:* neutral *Target:* Mujica *Attitude type:* negative
  - **Expressive subjective element** *Text anchor:* no se detalla nada *Source:* <writer> *Intensity:* high *Attitude type:* negative
  - **Expressive subjective element** *Text anchor:* nunca se concretará *Source:* <writer> *Intensity:* high *Attitude type:* negative

Se puede notar como en la primera parte de la conjunción la fuente es objetiva y en la segunda es subjetiva. Aquí el conector pero hace que la segunda parte sea la más importante para definir la subjetividad y la polaridad de la frase, como queda claro en el nuevo esquema.

A.6. Segunda etapa de anotación

En esta etapa nuevamente se procedió a anotar por separado otra tanda de opiniones de las mismas características pero con los cambios en el esquema de anotación explicados en la sección anterior y luego de haber unificado los criterios.

Al finalizar y comparar los resultados, se pudo notar un gran avance en cuanto a la disminución de las diferencias de anotación.

A.7. Conclusiones de la primera fase de anotación

La primera fase de anotación se considera exitosa, ya que se cumplieron los objetivos planteados de estudiar el problema y el corpus con el que se trabaja en este proyecto y conocerlo en profundidad.

Se encontraron problemas como las imperfecciones provenientes de BuscOpiniones para recuperar las oraciones bien formadas, las faltas de contexto y una concordancia entre anotadores menor a la esperada. Todos estos problemas encontrados serán profundizados en el resto de este capítulo.

Además, se descartó la utilización de algunos de los métodos estudiados en los trabajos relacionados, ya que se encontró que no se ajustan al problema planteado. Por ejemplo, se descartaron los que sugieren la utilización únicamente de los adjetivos para el análisis, ya que se anotaron otros tipos de palabras como sustantivos y conectores como fundamentales para conocer la polaridad.

Finalmente, se generó un esquema que puede ser de utilidad para realizar una anotación de sentimiento detallada. Sin embargo, se observó que el esquema es demasiado profundo y verboso como para permitir una anotación rápida, por lo que no es adecuado para la etapa de anotación de un corpus.