

UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA

PROYECTO DE GRADO

**Estudio de las opiniones utilizando
análisis de sentimientos a nivel de aspecto**

Autor:
Danilo Amaral

Supervisores:
Dr. Guillermo Moncecchi
Dra. Aiala Rosá



Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación

6 de mayo de 2020

«We can only see a short distance ahead, but we can see plenty there that needs to be done.»

Alan Mathison Turing, *Computing Machinery and Intelligence* (1950). p. 460.

Resumen

La generación de opiniones de forma masiva en medios sociales hace posible el diseño de técnicas de aprendizaje que permiten obtener conocimiento útil para la toma de decisiones de organizaciones y empresas. Un sistema capaz de inferir nueva información a partir de las opiniones públicas en las redes aporta valor al negocio y da al usuario la capacidad de procesar rápidamente información que realmente sea útil para un propósito específico.

En el presente documento se describe el trabajo realizado en el marco del proyecto de grado «Evolución de opiniones en redes sociales sobre temas de interés público». El mismo pretende llevar el estudio de la opinión pública al campo de la inteligencia artificial sobre el cual se aborda el problema de extracción automática de representaciones estructuradas de las opiniones de los usuarios contenidas en los textos. Se estudian los componentes y aspectos de una opinión con el objetivo de construir un sistema de aprendizaje automático para la identificación de aspectos sobre las opiniones emitidas por los usuarios.

Finalmente se implementa un sistema de aprendizaje que determine la polaridad de los aspectos involucrados en la opinión bajo un cierto tópico.

Índice general

Resumen	v
1. Introducción	1
1.1. Descripción del problema general	1
1.2. Objetivos del proyecto	3
1.3. Organización del documento	3
2. Marco Teórico	5
2.1. Niveles de profundidad en el análisis	6
2.2. Modelo de Opiniones	7
2.2.1. Definiciones	7
2.3. Tareas en la Minería de Opinión	11
2.4. Corpus	14
2.4.1. Conjunto de datos de entrenamiento de SemEval	14
2.4.2. Conjunto de datos de entrenamiento en TASS	15
2.5. Algoritmos de Aprendizaje	17
2.5.1. Redes Neuronales Artificiales	17
2.5.2. Representación de palabras en espacios de vectores	22
2.5.3. Redes Neuronales Convolucionales	25
2.6. Trabajos relacionados.	28
3. Descripción general de la solución	31
3.1. Arquitectura de la solución	31
3.2. Modelo de dominio	34
3.2.1. Traducción de corpus utilizados al modelo definido	36
3.3. Preprocesamiento de los textos de los documentos	41
3.4. Representación de las clases de salida	43
4. Algoritmos de aprendizaje y recursos utilizados	49
4.1. Algoritmos de clasificación	49
5. Evaluación del sistema propuesto	57
5.1. Métricas utilizadas	57
5.2. Análisis de los resultados	58
6. Conclusiones	65
A. Descripción de los recursos utilizados	67
A.0.1. Restaurant Reviews	67
A.0.2. Social-TV Corpus	71
A.0.3. STOMPOL	71

Índice de figuras

2.1. Comentario de ejemplo.	8
2.2. Tabla de relaciones para el corpus Restaurant Reviews	15
2.3. Tabla de relaciones para el corpus Laptops	15
2.4. Fragmento del corpus Social-TV 2.4.2	16
2.5. Fragmento del corpus STOMPOL 2.4.2	17
2.6. Vectores de palabras mediante Skip-ngram	23
2.7. Vectores de palabras mediante CBOW	24
2.8. Avance del filtro stride = 1	27
3.1. Arquitectura de la solución	32
3.2. Estructura del corpus de opiniones	34
3.3. Conjunto de relaciones para restaurant review	36
3.4. Fragmento del corpus Restaurant Reviews 2.4.1	37
3.5. Traducción sobre corpus 2.4.1	38
3.6. Fragmento del corpus STOMPOL formateado 2.4.2	40
3.7. Ejemplo ilustrativo: Fragmento de Restaurant Review formateado 3.5	44
3.8. Corpus para ABSA: Restaurant Reviews	44
3.9. 1. ^{er} método para ABSA: Binarización en 3.8	45
3.10. Corpus para ABSA: Restaurant Reviews	46
3.11. 2. ^{do} método para ABSA: BR para entidades en 3.10	46
3.12. 2. ^{do} método para ABSA: BR para aspectos en 3.10	46
3.13. 2. ^{do} método para ABSA: BR para la polaridad en 3.10	47
4.1. Conjunto de entrenamiento	49
4.2. 2. ^{do} método para ABSA: BR para aspectos en 3.10	50
4.3. Multi-Output CNN para ABSA	52
4.4. Recursos utilizados	53
4.5. Hiperparámetros en la red.	54
5.1. Identificación de aspectos para Restaurant Reviews 2.4.1	58
5.2. Identificación de aspectos para Restaurant Reviews 2.4.1 en inglés	59
5.3. Identificación de aspectos para Laptops 2.4.1	59
5.4. Análisis de sentimientos para Restaurant Reviews 2.4.1	60
5.5. Análisis de sentimientos para Restaurant Reviews 2.4.1 en inglés	61
5.6. Análisis de sentimientos para Laptops 2.4.1	61
5.7. Identificación de entidades para Social TV 2.4.2	62
5.8. Identificación de entidades para STOMPOL 2.4.2	63
5.9. Identificación de aspectos para STOMPOL 2.4.2	63
5.10. Análisis de sentimientos para Social TV 2.4.2	64
5.11. Análisis de sentimientos para STOMPOL 2.4.2	64

Capítulo 1

Introducción

Las opiniones son fundamentales para las actividades que requieren la toma de decisiones. Los políticos están interesados en las opiniones de sus potenciales votantes para tomar decisiones en su campaña política [56, 4, 16, 54, 85], los votantes se nutren de las opiniones de los diarios a modo de entender lo que las figuras públicas apoyan o se oponen [58, 77], las empresas estudian las opiniones de sus clientes para la evaluación de sus productos [65, 5] y los clientes buscan las opiniones de otros a la hora de elegir [67, 95].

El surgimiento de los medios sociales ha supuesto un gran avance en términos de cómo accedemos a nueva información; sin embargo, cada vez que alguien intenta buscar lo que otras personas piensan sobre algo, la respuesta es una enorme cantidad de datos que muchas veces solo dificulta la búsqueda de información útil para un propósito específico [17]. Es así que surge la necesidad de técnicas para la recuperación de información en base a las opiniones emitidas en medios sociales [28].

1.1. Descripción del problema general

La Minería de Opinión [27] es una disciplina de la lingüística computacional que estudia técnicas de procesamiento del lenguaje natural y minería de textos con el fin de extraer información subjetiva [61] a partir de contenidos de texto generados por distintos usuarios; para luego procesarlos y determinar la interpretación de los usuarios sobre ciertos tópicos. [41]

Distinguimos la Minería de Opinión del Análisis de Sentimiento (Sentiment Analysis), ya que el alcance de la polaridad (el sentimiento expresado por el autor) dentro de la opinión [43] es distinta. Mientras que en el análisis de sentimientos la polaridad abarca la opinión como un todo, en la minería de opinión recuperamos los componentes que conforman la opinión y valoramos aquellas características sobre las cuales el autor está expresando [41, 61, 45].

Por ejemplo, supongamos que en un blog de reseñas de celulares, un usuario realiza el siguiente comentario:

«Los celulares de gama alta tienen excelente resolución de pantalla y muy buena batería pero salen caros.»

El análisis de sentimientos a nivel de oración la clasifica positivamente, ya que en el texto aparecen palabras de carácter positivo como «excelente» y «muy bueno». Sin embargo, a nivel de opinión, este usuario anónimo estaría involucrando tres rasgos o aspectos en esta entidad denominada «celular» donde cada uno de estos es valorado de forma distinta.

Un sistema de minería de opinión debería recuperar esta información, entender que el objeto involucrado en la opinión son los celulares de gama alta como un todo, pero además, que el usuario está realizando una valoración sobre los rasgos de este objeto. Es decir, nuestro usuario (desconocido hasta el momento) valora positivamente a la resolución de pantalla y la batería, pero negativamente al precio de estos celulares. Este ejemplo es claro para entender también que el algoritmo utilizado por el sistema debería deducir que en el comentario se está hablando de un rasgo de forma implícita, el precio. En próximos capítulos estudiaremos los distintos niveles de alcance en el estudio de la opinión.

La información subjetiva [61] es aquella que no sólo contiene el punto de vista de la persona que la expone sino que también está influida, de alguna manera, por sus intereses y deseos personales. Ahí mismo es cuando la minería de opinión debe actuar, recuperando aquellas opiniones de usuarios que expresan algún tipo de emoción por el tema que se está hablando [43], digamos; entender el punto de vista del usuario sobre los distintos rasgos del objeto (o los objetos) involucrados en la opinión y a partir de ello, valorarlos en distintos niveles de intensidad según la emoción que el autor de la opinión expone.

1.2. **Objetivos del proyecto**

Dentro del marco del proyecto de grado pretendemos cumplir los requisitos resumidos en los siguientes puntos:

- Estudiar la minería de opinión en términos de entidades y aspectos.
- Diseñar un sistema de aprendizaje automático que realice las tareas definidas para el problema de minería de opiniones.
- Implementar el sistema y evaluar resultados.

1.3. **Organización del documento**

El resto del documento se subdivide en siete capítulos estructurados de la siguiente forma:

- El **Capítulo 2** desarrolla el marco teórico utilizado para la construcción de la solución y define las características del problema a resolver. Incluye una recopilación de las definiciones estudiadas y los distintos niveles de alcance en el análisis de opiniones.
- El **Capítulo 3** presenta el diseño de la solución propuesta. Contiene detalles de los componentes que conforman la arquitectura de la solución, los métodos utilizados para identificar entidades y aspectos sobre los textos de los documentos y la adaptación a los distintos corpus de la literatura.
- El **Capítulo 4** contiene detalles de implementación de los algoritmos. Presentamos los distintos métodos de aprendizaje utilizados en la construcción de la solución.
- En el **Capítulo 5** define las métricas utilizadas para la evaluación del sistema y un análisis de los resultados obtenidos.
- Finalmente, el **Capítulo 6** plantea las conclusiones y propuesta de trabajo a futuro.

Capítulo 2

Marco Teórico

El término «Minería de Opinión» aparece por primera vez en Dave et al. [17] en una conferencia sobre la web social (WWW2003) donde se propuso un sistema de clasificación semántico de reseñas de productos basado en opiniones de usuarios. Previamente en un reporte técnico de la sociedad de procesamiento de información en Japón (2001) [35] se desarrolla un motor de búsqueda para la extracción de información de opiniones en internet y se define el término «opinión» como un objeto de tres entidades: el nombre del objeto, el atributo y su evaluación.

Paralelamente, surge una tarea relacionada denominada Análisis de sentimientos que consiste en determinar el sentimiento asociado a los textos de los documentos. J. Yi et al. [31] sientan las bases del estudio y la definen mediante una tarea en dos pasos: la búsqueda de expresiones subjetivas para un objeto dado y la determinación de la polaridad de las expresiones. Sin embargo la investigación sobre sentimientos apareció mucho antes: en V. Hatzivassiloglou et al. [86] se implementó un sistema para predecir la orientación semántica de adjetivos en una investigación realizada para la octava Conferencia de la Asociación para la Lingüística computacional en 1997 (ACL 1997) y J. Wiebe et al. presentó en la Conferencia internacional de Lingüística Computacional de 1990 (COLING-1990) un sistema de identificación de oraciones subjetivas dentro de la narrativa [92].

Un trabajo importante en el área fue [63] presentado en la conferencia sobre métodos empíricos en 2002 en la Universidad de Pennsylvania por Pang, Lee y Vaithyanathan. Esta investigación se basó en la determinación del sentimiento sobre reseñas de películas y se concluye que las técnicas de Aprendizaje Automático mejoran el rendimiento de las líneas base generadas por los expertos humanos.

Los enfoques de hoy se basan en la obtención del sentimiento en distintos niveles de análisis. El nivel de documento clasifica el sentimiento en base a la reseña como un todo, el nivel de frase mediante lo expresado en cada oración y el de aspectos respecto a las características específicas de las entidades en los textos [53].

Los dos primeros enfoques resultaron incompletos ante la realidad de las empresas u organizaciones que quieren saber en detalle el comportamiento de sus productos [93]. Según B. Liu [42] ambos enfoques son insuficientes para descubrir las preferencias de los usuarios ya que existen más de una opinión en un mismo texto; por lo tanto, el análisis debe ser más específico.

En este trabajo nos enfocaremos en el último nivel de análisis denominado «Análisis de Sentimiento basado en aspectos» el cual permite recuperar el sentimiento

asociado a los objetos (o entidades) definidas en los textos de los documentos, lo que facilita el análisis de las opiniones emitidas por los usuarios. [45, 42].

2.1. Niveles de profundidad en el análisis

El análisis de sentimiento en los textos dentro de la minería de opinión se puede realizar en distintos niveles de profundidad. Cada uno de los cuales aborda el problema de clasificación de las opiniones en los textos desde una perspectiva distinta a la hora de determinar el alcance del sentimiento en la opinión.

Los siguientes son los distintos niveles de análisis para la clasificación de las opiniones en los textos de los documentos:

- **Análisis de sentimiento a nivel de documento.** Es la forma más simple de clasificación. Se considera el documento como la única unidad de información, es decir, el alcance de la opinión es el documento como un todo. Para que esta tarea tenga sentido, se supone que el documento tiene una única opinión sobre una sola entidad, donde además cada opinión es emitida por un solo autor (*holder*) en un tiempo dado. Este enfoque no es adecuado si el documento contiene opiniones sobre diferentes objetos. Obsérvese que en la práctica, si un documento contiene más de una entidad, el autor de la opinión puede evaluar como positivo sobre algunas y negativo sobre otras. Por lo tanto, no tiene sentido asignar una orientación de sentimiento a todo el documento. Tampoco tiene sentido si muchos autores expresan opiniones sobre un único documento por que cada uno lo puede evaluar desde distintos puntos de vistas.
- **Análisis de Sentimiento de nivel de oración.** En este nivel de análisis consideramos a cada oración como la unidad de información y cada una de ellas puede tener una opinión. Es decir, la polaridad se calcula para cada oración por separado. Esta tarea se puede resolver mediante dos problemas: el primero determina si la oración es una opinión o no, denominada clasificación de subjetividad [61, 43]. Mientras que el segundo es un problema de clasificación multiclase denominado análisis de sentimiento, el cual asigna una clase “positivo”, “negativo”, “neutral” a una oración. Obsérvese que este nivel es apropiado para oraciones simples que involucran una característica de un solo objeto, por ejemplo, «la calidad de imagen de esta cámara es increíble». Sin embargo, para oraciones compuestas y complejas, cabe la posibilidad de que se exprese más de un sentimiento.
- **Análisis de Sentimiento de nivel de aspecto (ABSA - Aspect-Based Sentiment Analysis).** Es el nivel de análisis más fino de los tres, centrado en la identificación de los aspectos relacionados con la entidad de estudio y luego en la determinación de la polaridad basada en cada aspecto de la entidad [28, 41, 43]. El análisis de sentimiento a nivel de documento o de oración no identifica el conjunto de objetivos de opinión que el texto está expresando. Es decir, clasifican la oración como un todo y no en base a los objetos en la misma. Incluso si asumimos que cada documento evalúa una entidad única, un documento de opinión positiva no significa que el autor tenga opiniones positivas sobre todos los aspectos de la entidad. Para un análisis más completo, necesitamos descubrir los aspectos y determinar si el sentimiento es positivo o negativo en cada aspecto. Por eso este marco es clave para el análisis de opiniones y será nuestro enfoque en este documento.

2.2. Modelo de Opiniones

Esta sección se encarga de exponer la terminología utilizada en el documento que corresponden a definiciones obtenidas mayoritariamente de B. Liu et al. [41, 40, 6, 44, 43, 45, 42] y B. Pang et al. [63, 62, 61].

2.2.1. Definiciones

Definición 2.2.1 (Opinión).

Una *opinión* O es un *quíntupla* (e, a, s, h, t) , donde:

- e es el nombre de una entidad.
- a es un aspecto de e .
- s es la orientación de la opinión sobre el aspecto a de la entidad e .
- h es el titular de la opinión.
- t es el momento en que la opinión es expresada por h .

Usualmente la orientación s se expresa mediante la escala de valores: positivo, negativo y neutral, denominándose orientaciones de sentimiento (o polaridades) [43, 42].

El par $ot = (e, a)$ es usualmente llamado **objetivo de la opinión** (*opinion target*, OT) ya que junto con el texto es la información suficiente para obtener la orientación de la opinión.

A continuación realizamos una serie de observaciones de la definición anterior [43, 42]:

1. Una opinión consta de dos componentes clave: El objeto evaluado en la opinión y el sentimiento asociado. Es decir el núcleo de la opinión se define mediante el par (ot, s) , donde ot se corresponde con lo evaluado en la opinión, y s un sentimiento definido mediante una escala de valores o una puntuación numérica que expresa la intensidad (por ejemplo, 1 a 5 estrellas en comentarios de restaurantes o «me gusta» y «no me gusta» en redes sociales).
2. Las cinco piezas de información en la *quíntupla* deben corresponderse de la siguiente manera: la opinión O debe ser emitida por el titular de opinión h sobre el aspecto a de la entidad e en el momento t . Cualquier otra combinación es errónea.
3. Los cinco componentes son esenciales. La falta de alguno genera problemas de inconsistencias. Por ejemplo, si el componente del tiempo no fue recuperado, no podremos analizar opiniones sobre una entidad en el tiempo, lo que a menudo es muy importante en la práctica porque una opinión de hace dos años y una de ayer no es la misma.
4. Debido a la simplificación en dos niveles de representación, tenemos pérdida de información representable en el texto.

Por ejemplo, supongamos que un usuario escribió «La tinta de esta impresora es cara». Si a uno no le importa ningún atributo de la tinta, esta oración simplemente da una opinión negativa sobre la tinta, que es un aspecto de la

entidad «impresora». Sin embargo, si queremos estudiar opiniones sobre diferentes aspectos de la tinta, por ejemplo, el precio y la calidad, la tinta debe tratarse como una entidad separada. Por lo tanto, a pesar de que este comentario puede ser modelado con la definición de opinión; la relación entre las partes de la impresora debe ser guardada por separado. La definición cubre la información esencial de una opinión que es suficiente para la mayoría de las aplicaciones. Una definición demasiado compleja puede hacer que el problema sea extremadamente difícil de resolver [6].

5. El titular de una opinión (*opinion holder*) h es la persona u organización que expresa la opinión [36, 91, 8, 12].
6. Esta definición proporciona un marco para transformar texto no estructurado en datos estructurados. La quintupla anterior es básicamente un esquema de base de datos, en función del cual las opiniones extraídas se pueden colocar en una tabla. Luego, se puede realizar un amplio conjunto de análisis de opiniones cualitativos, cuantitativos y de tendencias utilizando herramientas más sofisticadas.
7. La opinión definida aquí es solo un tipo de opinión, llamada opinión regular. Otro tipo es la opinión comparativa [32, 33]. A modo de simplificación, nos enfocaremos en opiniones regulares y simplemente los llamaremos opiniones.

Vamos a utilizar el comentario de ejemplo 2.1 definido en [6] para mostrar estos conceptos. Se incluye un número asociado a cada oración a modo de identificarla para referencias posteriores.

Publicado por:	John Smith
Fecha:	10 de septiembre de 2011
Comentario:	<ol style="list-style-type: none"> 1. Hace seis meses compré una cámara Canon G12. 2. Simplemente me encanta. 3. La calidad de la imagen es increíble. 4. La duración de la batería también es larga. 5. Sin embargo, mi esposa piensa que es demasiado pesada para ella.

FIGURA 2.1: Comentario de ejemplo.

A partir del comentario anterior se realizan las siguientes observaciones:

- La reseña tiene una serie de opiniones, tanto positivas como negativas, sobre la cámara Canon G12. La oración (2) expresa una opinión positiva sobre la cámara Canon en su conjunto. La oración (3) expresa una opinión positiva sobre la calidad de imagen, la oración (4) sobre la duración de la batería. Sin embargo la oración (5) expresa una opinión negativa sobre el peso de la cámara [6].
- El objetivo de la opinión en la oración (2) es Canon G12, y el objetivo de la opinión en la oración (3) es la calidad de imagen de Canon G12. Obsérvese que se corresponden con sujetos gramaticales distintos lo que hace difícil su tratamiento. El objetivo también se llama tópico (topic) en la literatura.
- Esta revisión tiene opiniones de dos personas, que se denominan fuentes, poseedores o titulares de opinión [36, 91]. El titular de las opiniones en las oraciones (2), (3) y (4) es el autor del comentario ("John Smith"), pero para la oración (5) es la esposa del autor.
- La fecha de la revisión es el 10 de septiembre de 2011. Este dato es importante en la práctica porque a menudo se quiere saber cómo cambian las opiniones con el tiempo [6].

Obsérvese que en la práctica la descripción completa de e es difícil de definir, pero además puede que ni siquiera aparezca en la misma oración. Por ejemplo, en (3), el objetivo de opinión es en realidad «calidad de imagen de la Canon G12», pero menciona solo «calidad de imagen» ya que se deduce del contexto que se habla de la cámara [27, 40, 43, 42].

Vamos a definir e de manera estructurada con múltiples niveles lo que facilita enormemente la extracción de opiniones y el uso posterior de los resultados de la opinión extraída. Por ejemplo, la «calidad de imagen de Canon G12» se puede descomponer en una entidad, un atributo de la entidad y representarse como un par: (Canon-G12, calidad de imagen) [27, 40, 43, 42].

Usemos el término entidad para denotar el objeto de la opinión objetivo que se ha evaluado. La entidad se puede definir de la siguiente manera [27, 40, 43].

Definición 2.2.2 (Entidad).

Una **entidad** (*entity*) e es un producto, servicio, persona, evento, organización o tema.

Lo representamos como un par $e: (T, W)$ donde:

- T es una jerarquía de componentes (o partes), sub-componentes, etc.,
- W es un conjunto de atributos de e .

Ejemplo 2.2.1.

Un modelo particular de cámara es una entidad, por ejemplo, Canon G12. Tiene un conjunto de atributos: calidad de imagen, tamaño y peso; y un conjunto de partes: lente, visor y batería [43]. La batería también tiene su propio conjunto de atributos: la duración de la batería y el peso de la batería. Un tema también puede ser una entidad, por ejemplo, aumento de impuestos, con sus partes «aumento de impuestos para los pobres», «aumento de impuestos para la clase media» y «aumento de impuestos para los ricos» [43].

Esta definición esencialmente nos describe la entidad como una descomposición jerárquica basada en la relación de sus partes. El nodo raíz es el nombre de la entidad, por ejemplo, *Canon G12* en el comentario anterior. Todos los demás nodos son partes y subpartes, etc. Se puede expresar una opinión sobre cualquier nodo y cualquier atributo del mismo [43].

Ejemplo 2.2.2.

En en el comentario anterior, la oración (2) expresa una opinión positiva sobre la entidad Canon G12 como una cámara en su conjunto. La oración (3) expresa una opinión positiva sobre el atributo de calidad de imagen de la cámara [43]. Claramente, también se pueden expresar opiniones sobre partes o componentes de la cámara.

La recuperación de entidades en los textos utilizando esta definición es una tarea difícil de modelar [43, 42] ya que necesita para su representación en texto plano una estructura demasiado compleja para llevarla a la práctica. Pero además, la mayoría de las aplicaciones no requieren un nivel de detalle tan específico.

Es así que, a efectos de simplificar nuestro problema y adaptarlo a la práctica utilizaremos una jerarquía en dos niveles. Emplearemos el término **aspecto** para denotar tanto la distintas partes como atributos contenidos en la entidad [43, 42].

Definición 2.2.3 (Aspecto).

*Los **aspectos** (aspects) de una entidad e son los componentes y atributos de e.*

Cuando una opinión es sobre la entidad en sí misma como un todo, el aspecto especial GENERAL se usa para denotarla [43, 42].

2.3. Tareas en la Minería de Opinión

Visto el marco teórico de la sección anterior, ahora podemos presentar el objetivo y las tareas clave del análisis de opiniones [40, 61, 43]. Recuperar para todo texto en el corpus el conjunto de todas las quintuplas de opinión $O = (e, a, s, h, t)$ [6, 44, 43]. Las tareas clave se derivan de los cinco componentes de la quintupla:

- 1. Categorización e identificación de expresiones de entidad.**
Dado el texto de la opinión, recuperar de forma automática las expresiones de entidades en los textos, identificar aquellas que refieran a la misma entidad y agrupar las mismas en categorías.
- 2. Extracción de señales de aspectos y su categorización a partir de las entidades.**
Dado el texto de la opinión y sus entidades, obtener para cada entidad las expresiones de los aspectos en los textos y agruparlas en aspectos que refieran a lo mismo.
- 3. Determinación de la polaridad usando análisis de sentimiento basado en los aspectos de las entidades de la opinión.**
Determinar, para cada *opinion target* ot , construido en base a las entidades y aspectos extraídos, la orientación del sentimiento s utilizando técnicas de análisis de sentimiento basado en aspectos a nivel de oración.
- 4. Recuperación del titular de la opinión.**
Recuperación de información para la obtención del titular de la información a partir de los textos en los documentos. En el marco de este proyecto, simplificamos el problema asumiendo que el autor de la opinión es aquel que realiza el comentario en el medio social. Es decir, el autor de la opinión es un usuario totalmente identificado por el sistema.
- 5. Obtener el momento de la opinión.**
Recuperación de información para la obtención del momento por el cual la opinión fue expresada. Asumimos que el sistema almacena el tiempo en el cual la opinión es ingresada al sistema.
- 6. Generación de las quintuplas de opinión.**
Producir todas de opiniones (e, a, s, h, t) en el documento en función de los resultados de las tareas anteriores. Esta tarea que a priori parece simple en la realidad resulta ser compleja como lo vamos a ver en el próximo ejemplo. Obsérvese que en el marco de este proyecto, necesitamos producir simplemente las ternas (e, a, s) ya que el autor y el tiempo son conocidos.

La primera tarea se enfoca al primer componente de la quintupla, la entidad. Necesitamos reconocer y extraer entidades en los textos. La tarea es similar al reconocimiento de entidades nombradas (NER) en la extracción de información definida en [26, 59, 75].

Después de la extracción, tenemos que clasificar las entidades extraídas. En el texto las personas escriben la misma entidad de diferentes maneras. Por ejemplo, Motorola puede escribirse como Mot, Moto y Motorola. Necesitamos reconocer que todos se refieren. La palabra o frase en el texto de la opinión que hace referencia a una entidad se denomina **expresión o señal de entidad** (*entity cue*) y el proceso de agrupar expresiones de entidades en nombres de entidades se denomina **categorización de entidades**.

La segunda tarea se enfoca al segundo componente, el aspecto. El problema es básicamente el mismo que para las entidades. Por ejemplo, imagen y foto son el mismo aspecto para las cámaras. Una **expresión o señal de aspecto** (*aspect cue*) es una palabra o frase en el texto que indica la presencia de un aspecto. Cada aspecto también debe tener un nombre único en una aplicación en particular. El proceso de agrupar expresiones de aspecto en nombres de aspectos se denomina **categorización de aspecto**.

Las expresiones de aspecto suelen ser sustantivos y frases nominales, pero también pueden ser verbos, sintagmas verbales, adjetivos y adverbios. Las siguientes definiciones definen dos tipos de expresiones de aspectos [27]:

- *Expresión de aspecto explícita*. Las expresiones de aspecto que son sustantivos y sintagmas nominales se denominan **expresiones de aspecto explícitas**. Por ejemplo, «calidad de imagen» en «La calidad de imagen de esta cámara es excelente» es una expresión de aspecto explícito.
- *Expresión de aspecto implícito*. Las expresiones de aspecto que no son sustantivos o frases nominales se denominan **expresiones de aspecto implícitas**. Por ejemplo, «cara» es una expresión de aspecto implícita en «Esta cámara es cara». La misma, implica el precio del aspecto.

Muchas expresiones de aspectos implícitos son adjetivos y adverbios que se usan para describir o calificar algunos aspectos específicos, por ejemplo, «costosos» (*precio*) y «confiablemente» (*confiabilidad*). También pueden ser verbos y sintagmas verbales, por ejemplo, «Puedo instalar el software fácilmente». «Instalar» indica el aspecto de la instalación.

El tercer componente en la definición de opinión es el sentimiento. La tercer tarea clasifica si el sentimiento sobre el aspecto es positivo, negativo o neutral. Los componentes cuarto y quinto son el titular de la opinión y tiempo respectivamente.

La minería de opinión basado en este marco 2.2.1 a menudo se denomina **análisis de sentimiento basado en aspectos** (o basado en características) [61, 44, 45].

Utilicemos un ejemplo para ilustrar los resultados del análisis.

Publicado por:	big John
Fecha:	15 de septiembre de 2011
Comentario:	<ol style="list-style-type: none"> 1. Compré una cámara Samsung y mis amigos trajeron una Canon ayer. 2. En la semana pasada, los dos usamos mucho las cámaras. 3. Las fotos de mi Samy no son tan buenas, y la duración de la batería es corta. 4. Mi amigo está muy feliz con su cámara y le encanta su calidad de imagen. 5. Quiero una cámara que pueda tomar buenas fotos. 6. Lo voy a devolver mañana.

1. En la extracción de entidades se debe recuperar las expresiones de entidad, "Samsung", "Samy" y "Canon", agrupando cada una en una entidad con nombre ya que representan la misma entidad. Por ejemplo, SAMSUNG y CANON.
2. Luego, se debe extraer expresiones de aspecto «imagen», «foto» y «duración de la batería», y agrupar las primeras dos ya que para las cámaras, ambos se corresponden con el mismo componente.
3. La siguiente tarea debe recuperar que el titular de las opiniones. Se deduce que el autor del blog es quien realiza todas las opiniones, salvo la oración número (4) quien es el amigo del autor.
4. La próxima tarea debe encontrar el momento en que se publicó el blog. Se encuentra en el campo de la fecha, en este caso.
5. La penúltima tarea debe clasificar que la oración (3) da una opinión negativa a la calidad de imagen de la cámara Samsung y también una opinión negativa a la duración de la batería. La oración (4) da una opinión positiva a la cámara Canon en su conjunto y también a su calidad de imagen. La oración (5) aparentemente expresa una opinión positiva, pero no lo hace. Es decir el sistema también requiere identificar expresiones lingüísticas complejas (modalidad).
6. La última tarea consiste en generar las quintuplas de opinión, necesitamos saber a qué se refieren «su cámara» y «su» en la oración (4). Esta tarea es de alta complejidad y requiere un sistema de análisis complejo que por ejemplo realice correlaciones e identifique anáforas. Finalmente podemos generar las siguientes cuatro quintuplas de opinión:
 - a) (Samsung, picture_quality, negativo, big John, septiembre 15-2011)
 - b) (Samsung, battery_life, negativo, big John, septiembre 15-2011)
 - c) (Canon, GENERAL, positivo, big John's_friend, septiembre 15-2011)
 - d) (Canon, picture_quality, positive, big John's_friend, septiembre 15-2011)

2.4. Corpus

La siguiente sección presenta diferentes corpus utilizados para el Análisis de sentimientos basados en aspecto.

2.4.1. Conjunto de datos de entrenamiento de SemEval

SemEval ¹(*International Workshop on Semantic Evaluation*) es una serie de evaluaciones de modelos semánticos organizada por la Asociación de Lingüística Computacional (*Association for Computational Linguistics, ACL*) ², sociedad que organiza las principales conferencias de PLN desde 1998.

La tarea de análisis de sentimiento basado en aspectos (*Aspect Based Sentiment Analysis, ABSA*), fue la cuarta tarea introducida en 2014 [48] y tuvo como objetivo realizar la extracción y el análisis de aspectos sobre comentarios de restaurantes en inglés. Esta tarea fue redefinida en SemEval-2015 [47] basándose en la anterior y consolidando las tareas en el marco de la minería de opinión visto en 2.2.1. Finalmente en SemEval-2016 [46] se incluye anotaciones a nivel de oración en los corpus así como múltiples dominios incluidos restaurantes, computadoras portátiles y en diferentes idiomas.

A continuación presentamos la semántica del corpus de restaurantes presentado en la SemEval-2016 con sus distintas anotaciones ³ [46]. Destacamos que los demás corpus siguen la misma estructura pero claramente cambian la semántica de sus anotaciones dados los distintos temas.

Corpus Restaurant Reviews

Restaurant Reviews es un corpus de reseñas de Restaurantes de la SemEval en su edición 2015 para la tarea número 12 conocida como Aspect Based Sentiment Analysis (ABSA) [47], posteriormente fue utilizado en SemEval-2016 para la tarea 5 [46] en versiones en español e inglés. El objetivo era identificar opiniones expresadas sobre reseñas de restaurantes hacía entidades específicas definidas y sus atributos [48]. En esta edición una entidad, como objetivo de evaluación, puede ser un restaurante como un todo (por ejemplo, «el restaurante», «La costa de oro» (viejo restaurante famoso en la avenida 18 de Julio de Montevideo)), su ambiente, el lugar donde está ubicado, la comida y tragos que ofrece, etc.

Las entidades son representadas mediante anotaciones (etiquetas) de la forma $\{E\#A, OTE, P\}$ asignadas a nivel de oración en las reseñas teniendo en cuenta el contexto de la misma. La tabla 2.2 indica cómo se deben corresponder las entidades y los atributos para formar la etiqueta E#A.

¹https://aclweb.org/aclwiki/SemEval_Portal

²<https://www.aclweb.org/>

³SemEval 2016 Task12. Aspect Based Sentiment Analysis (ABSA-16). Annotation Guidelines.http://alt.qcri.org/semeval2016/task5/data/uploads/absa2016_annotationguidelines.pdf

Categories Attributes	GENERAL	PRICES	QUALITY	STYLE&OPTIONS	MISCELLANEOUS
RESTAURANT	✓	✓	✗	✗	✓
FOOD	✗	✓	✓	✓	✗
DRINKS	✗	✓	✓	✓	✗
AMBIENCE	✓	✗	✗	✗	✗
SERVICE	✓	✗	✗	✗	✗
LOCATION	✓	✗	✗	✗	✗

FIGURA 2.2: Tabla de relaciones para el corpus Restaurant Reviews

Es así que dado una entidad, por ejemplo “RESTAURANT”, tiene el siguiente conjunto de aspectos { “GENERAL”, “PRICES”, “MISCELLANEOUS” }. Entonces por ejemplo, cuando se hace referencia en el texto al restaurante en forma general se lo etiqueta con “RESTAURANT#GENERAL”.

La expresión objetivo de la opinión (OTE) refiere a una mención en el texto de la entidad “E” evaluada en el par “E#A” y “P” es la polaridad asociada al par (E,A) mediante la escala de valores { “positive”, “negative”, “neutral” }. Por ejemplo, «La costa de oro fue el mejor de 18 de Julio.» es anotado con las siguientes etiquetas {RESTAURANT#GENERAL, La costa de oro, positive}.

Corpus Laptops

Laptops es un corpus de reseñas de computadoras personales para SemEval. Al igual que Restaurant Reviews el objetivo era identificar opiniones expresadas sobre reseñas de productos hacia entidades específicas definidas y sus atributos.

Categories Attributes	GENERAL	CONNECTIVITY	DESIGN_FEATURES	OPERATION_PERFORMANCE	MISCELLANEOUS	PORTABILITY	PRICE	QUALITY	USABILITY
BATTERY	✓	✓	✓	✓	✓	✓	✓	✓	✓
COMPANY	✓	✓	✓	✓	✓	✓	✓	✓	✓
CPU	✓	✓	✓	✓	✓	✓	✓	✓	✓
DISPLAY	✓	✓	✓	✓	✓	✓	✓	✓	✓
FANS_COOLING	✓	✓	✓	✓	✓	✓	✓	✓	✓
GRAPHICS	✓	✓	✓	✓	✓	✓	✓	✓	✓
HARDWARE	✓	✓	✓	✓	✓	✓	✓	✓	✓
HARD_DISC	✓	✓	✓	✓	✓	✓	✓	✓	✓
KEYBOARD	✓	✓	✓	✓	✓	✓	✓	✓	✓
LAPTOP	✓	✓	✓	✓	✓	✓	✓	✓	✓
MEMORY	✓	✓	✓	✓	✓	✓	✓	✓	✓
MOTHERBOARD	✓	✓	✓	✓	✓	✓	✓	✓	✓
MOUSE	✓	✓	✓	✓	✓	✓	✓	✓	✓
MULTIMEDIA_DEVICES	✓	✓	✓	✓	✓	✓	✓	✓	✓
OPTICAL_DRIVES	✓	✓	✓	✓	✓	✓	✓	✓	✓
OS	✓	✓	✓	✓	✓	✓	✓	✓	✓
PORTS	✓	✓	✓	✓	✓	✓	✓	✓	✓
POWER_SUPPLY	✓	✓	✓	✓	✓	✓	✓	✓	✓
SHIPPING	✓	✓	✓	✓	✓	✓	✓	✓	✓
SOFTWARE	✓	✓	✓	✓	✓	✓	✓	✓	✓
SUPPORT	✓	✓	✓	✓	✓	✓	✓	✓	✓
WARRANTY	✓	✓	✓	✓	✓	✓	✓	✓	✓

FIGURA 2.3: Tabla de relaciones para el corpus Laptops

Entonces, por ejemplo “super fast processor and really nice graphics” es anotado con las siguientes etiquetas {GRAPHICS#GENERAL, nice graphics, positive} y {CPU#OPERATION_PERFORMANCE, super fast, positive}.

2.4.2. Conjunto de datos de entrenamiento en TASS

El Taller de Análisis Semántico (TASS) (previamente, Taller de evaluación sobre Análisis de Sentimientos)¹ se ha venido realizando en el Congreso Internacional de la Sociedad Española de Procesamiento del Lenguaje Natural (SEPLN) desde el año 2012 con el fin de promover la investigación de análisis de opiniones en español. TASS fue el primer taller de evaluación de sistemas de análisis de opiniones escritas en español. Los siguientes son ejemplos de corpus utilizados.

¹<http://www.sepln.org/workshops/tass/>

Social-TV Corpus - TASS 2014 (SEPLN)

Social-TV ¹ es un corpus presentado en la TASS de la SEPLN construido durante la final de la Copa del Rey entre Real Madrid y Barcelona el 16 de Abril del 2014 en el estadio de Mestalla en Valencia. Se recopilieron más de 1 millón de tweets desde 15 minutos antes hasta 15 minutos después del partido. Los tweets irrelevantes se filtraron y se seleccionó un subconjunto de 2.773.

El corpus de Social-TV se dividió aleatoriamente en dos conjuntos: entrenamiento (1,773 tweets) y prueba (1,000 tweets), con una distribución similar de aspectos y sentimientos. El corpus de prueba se proporcionará sin ninguna anotación y se utilizará para evaluar los resultados proporcionados por los diferentes sistemas. La figura 2.4 ilustra un fragmento de este corpus.

```

1 <tweet id="456544898791907328">
2   <sentiment aspect="Equipo-Real_Madrid"
3     polarity="P">#HalaMadrid</sentiment> ganamos sin
4   <sentiment aspect="Jugador-Cristiano_Ronaldo"
5     polarity="NEU">Cristiano</sentiment>. .perdéis con
6   <sentiment aspect="Jugador-Lionel_Messi"
7     polarity="N">Messi</sentiment>. Hala
8   <sentiment aspect="Equipo-Real_Madrid"
9     polarity="P">Madrid</sentiment>! !!!!!
10
11 </tweet>
12 <tweet id="456544890550099968">
13   No digáis ahora que vaya robo, cuando el
14   <sentiment aspect="Equipo-Real_Madrid"
15     polarity="P">Madrid</sentiment>
16   ha sido superior claramente, y a
17   <sentiment aspect="Jugador-Lionel_Messi"
18     polarity="N">messi</sentiment> ni se le ha visto.
19 </tweet>

```

FIGURA 2.4: Fragmento del corpus Social-TV 2.4.2

Sin entrar en detalles sobre las categorías definidas en los mismos, podemos notar que la anotación tiene la forma "Categoría-Entidad", sin hacer un mención sobre el aspecto de la entidad, y que es anotada dentro del texto de la reseña, indicando la expresión de entidad contenida en el. Además la polaridad del sentimiento se anotó desde el punto de vista del usuario de Twitter, utilizando 3 etiquetas: {"P", "N", "NEU"}; no se hace distinción en los casos en que el autor no expresa ningún sentimiento o expresa un sentimiento no positivo ni negativo.

¹<http://www.sepln.org/workshops/tass/2017/>

STOMPOL - TASS 2015 (SEPLN)

STOMPOL (Spanish Tweets for Opinion Mining about POLitics)¹ es un corpus de tweets en español para el análisis de sentimientos a nivel de aspectos. El mismo está basado en la campaña política del 2015 para las elecciones locales en España donde sus opiniones se recuperaron los días 23 y 24 de Abril. El corpus está compuesto por 1.284 tweets y se ha dividido en un conjunto de entrenamiento (784 tweets), que se proporciona para construir los sistemas, y un conjunto de prueba (500 tweets) que se utilizarán para la evaluación. La figura 2.5 ilustra este corpus:

```

1 <tweet id="591267548311769088">
2   @ahorapodemos @Pablo_Iglesias_ @SextaNocheTV
3   Que alguien pregunte si habrá cambios en las
4   <sentiment aspect="Educacion"
5     entity="Podemos"
6     polarity="NEU">becas</sentiment> MEC
7     para universitarios , por favor.
8 </tweet>
9 <tweet id="591192167944736769">
10  #Arroyomolinos lo que le interesa al ciudadano son Políticos
11  cercanos que se interesen y preocupen por sus problemas
12  <sentiment aspect="Propio_partido"
13    entity="Union_Progreso_y_Democracia"
14    polarity="P">@UPyD</sentiment> VECINOS COMO TU
15 </tweet>

```

FIGURA 2.5: Fragmento del corpus STOMPOL 2.4.2

Cada tweet en el corpus fue anotado manualmente por dos anotadores diferentes, más un tercero en caso de desacuerdo, con la polaridad del sentimiento a nivel de aspecto. La polaridad del sentimiento se anotó desde el punto de vista del usuario de Twitter, utilizando 3 niveles: {"P", "N", "NEU"}. No se hace diferencia entre ningún sentimiento y un sentimiento neutral (ni positivo ni negativo). Cada aspecto político está vinculado a su correspondiente partido político y su polaridad.

Para más información acerca de la semántica de las etiquetas en los corpus presentados en esta sección, referirse al apéndice A.

2.5. Algoritmos de Aprendizaje

La próxima sección realiza una introducción a las redes neuronales artificiales; base que consideramos necesaria para comprender el resto del capítulo. Luego basado en estos principios, definimos las redes neuronales convolucionales y utilizando estos dos conceptos presentamos la arquitectura del sistema.

2.5.1. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (ANN - Artificial Neural Network) fueron propuestas por McCulloch y Pitts [52] en 1943 inspirado en una simulación abstracta de los sistemas nerviosos centrales, donde, a grandes rasgos, los mismos están constituidos por un conjunto de unidades llamados neuronas o nodos interconectados

¹<http://www.sepln.org/workshops/tass/2017/>

unos con los otros.

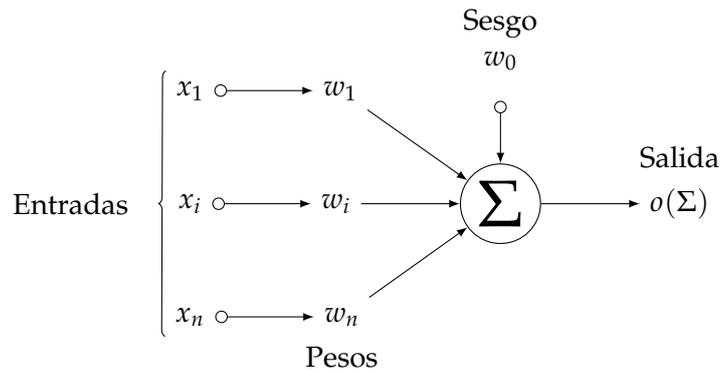
Luego, Frank Rosenblatt en 1958 [72] define la arquitectura para un modelo binario donde cada neurona era definida mediante una función denominada perceptrón que tenía un escalón o un umbral prefijado y utilizaba un conjunto de datos y algoritmo totalmente definido para ajustar los parámetros de la neurona al conjunto de datos. A continuación, definimos el perceptrón mediante,

Definición 2.5.1 (Perceptrón, Frank Rosenblatt, 1958).

Definimos el **Perceptrón simple** (o Adaline) como la unidad más sencilla de red neuronal artificial donde dado una secuencia de entradas de números reales $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$, produce una salida definida mediante:

$$o(x) = \begin{cases} 1 & , \text{ si } w_0 + \sum_{i=1}^n w_i x_i > 0 \\ -1 & , \text{ en otro caso.} \end{cases}$$

Donde $w_i \in \mathbb{R}, i = 0 \dots n$ valor real o peso que determina cuánto contribuye x_i al perceptrón.



Observaciones

1. $-w_0$ es el umbral, usualmente llamado sesgo (bias) que tiene que pasar el perceptrón para que retorne positivo, es decir

$$o(x) = 1 \Leftrightarrow \sum_{i=1}^n w_i x_i > -w_0$$

2. Si definimos $\vec{x} = (x_0, x_1, \dots, x_n)$, $\vec{w} = (w_0, w_1, \dots, w_n) \in \mathbb{R}^{n+1}$, luego la función perceptrón puede ser definida mediante,

$$o(x) = \text{sgn}(\vec{w}^T \vec{x})$$

w son los pesos que debe ajustar el perceptrón para que las entradas están contenidas en clases C_0 y C_1 linealmente separables (en lados opuestos de un hiperplano).

Para ajustar los pesos utilizamos un conjunto de entrenamiento

$$D = \{(\vec{x}_1, t_1), \dots, (\vec{x}_k, t_k)\}$$

Compuesto por k pares de vectores $n + 1$ dimensionales. La idea es ajustar D al perceptrón, es decir, ajustar \vec{w} para que

$$D \equiv \{(\vec{x}_1, o(\vec{x}_1)), \dots, (\vec{x}_k, o(\vec{x}_k))\}$$

Y una función de costo (loss function) E que representa una medida de error entre la salida del perceptrón y los valores reales, por ejemplo el error cuadrático medio

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2, \text{ donde } d = (\vec{x}_d, t_d) \in D, o_d = \vec{w}^T \vec{x}_d$$

Luego, la idea es buscar \vec{w} que minimiza $E(\vec{w})$, es decir,

$$\underset{\vec{w}}{\text{argmin}} E(\vec{w})$$

Los algoritmos que ajustan los pesos minimizando el gradiente de la función de costo se llaman algoritmos por descenso de gradiente (*Gradient Descent*) ya que iterativamente actualizan los pesos utilizando la dirección de la función de costo que produce el descenso más pronunciado $-\alpha \nabla E(\vec{w}_i) = -\alpha \left(\frac{\partial E}{\partial w_i} \right)_{i=1 \dots n}$.

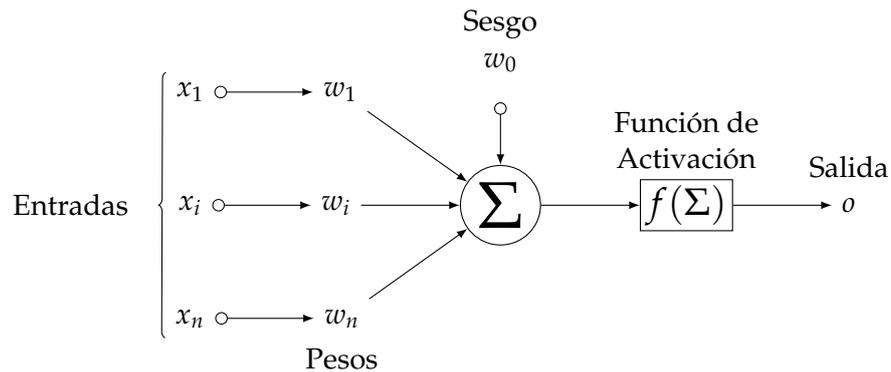
Obsérvese que el umbral del perceptrón es discontinuo e indefinido para el descenso por gradiente, justamente por eso la función signo no aparece en la definición de E . Para solucionar este problema definimos la unidad con umbral diferenciable de la siguiente manera:

Definición 2.5.2.

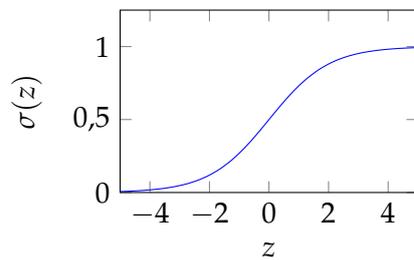
Definimos el **Perceptrón con umbral** como la unidad de red neuronal artificial donde dado una secuencia de entradas de números reales $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, produce una salida definida mediante,

$$o(x) = f\left(w_0 + \sum_{i=1}^n w_i x_i\right)$$

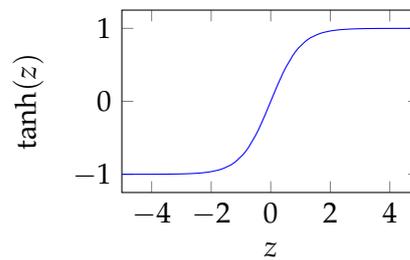
Donde $w_i \in \mathbb{R}, i = 0 \dots n$ son los pesos del perceptrón y f denominada función de activación.



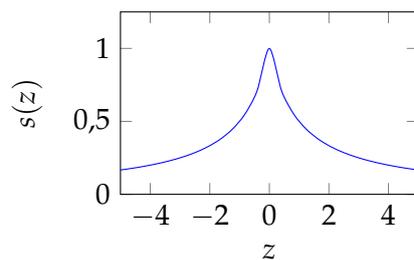
Usualmente f es una función no lineal, continua, diferenciable y monótona no decreciente [71]. Las siguientes son algunos ejemplos de funciones de activación:



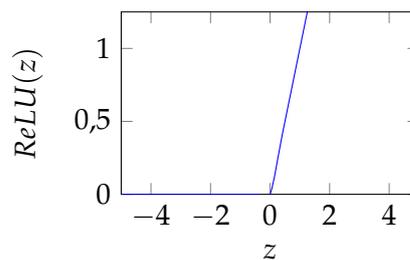
(a) Función logística (sigmoide).



(b) Tangente hiperbólica.



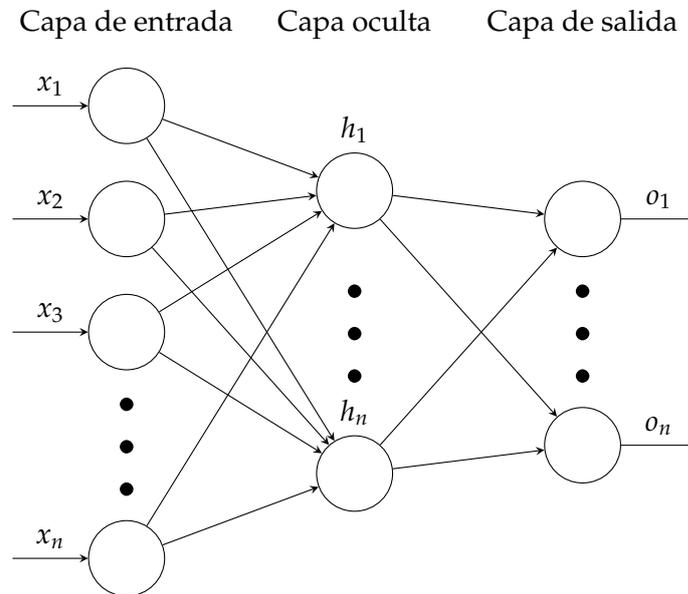
(c) Softsign.



(d) Unidad lineal rectificada (ReLU).

Perceptrón multicapa

El perceptrón multicapa [89, 73] es una red neuronal artificial compuesta por múltiples capas interconectadas de perceptrones con umbral, donde la salida de la capa anterior sirve de entrada a la próxima capa. Estas pueden clasificarse en tres tipos:



- **Capa de entrada (Input Layer).** Es la capa conformada por los patrones de entrada que dan señales al resto de la red. Los mismos no son objetivo de entrada de ninguna neurona en la red.
- **Capas ocultas (Hidden Layers).** Constituida por neuronas alimentadas por capas anteriores y cuyas salidas pasan a capas posteriores.
- **Capa de salida (Output Layer).** Neuronas cuyo valor de salida están en la salida de toda la red.

Una forma de entrenar estas redes es mediante un algoritmo por descenso de gradiente denominado **Backpropagation** [89, 73], el cual consiste en presentarle un patrón de entrada a la red, estimular la red enviando las señales a la capa de salida y luego estimar el error de la capa de salida utilizando la regla de la cadena.

2.5.2. Representación de palabras en espacios de vectores

La forma tradicional en que representamos las palabras como vectores era mediante *one-hot encoding* que se basaba en representar las palabras mediante un vector binario, cuyas coordenadas representan la lista de palabras en el vocabulario.

Por ejemplo, *mujer* y *reina* se representaban mediante $x_{\text{mujer}} = (0, \dots, 010, \dots, 0)$ y $x_{\text{reina}} = (0, \dots, 100, \dots, 0)$ donde solamente está la unidad en la coordenada que le corresponde a la palabra. Esta representación no solamente era simple de implementar sino que permitía rápidamente llevarla a un algoritmo de clasificación para el análisis de los textos. Sin embargo, tiene dos problemas muy grandes [55, 84]:

1. Las representaciones de los textos de palabras generaban una matriz dispersa, lo que dificulta la clasificación.
2. La representación por vectores binarios no capturan relaciones entre las palabras.

La representación mediante vectores de palabras (Word Embedding), es una técnica de ingeniería de atributos (Feature Engineering) para encontrar un modelo de lenguaje que transforme palabras de un vocabulario dentro de un espacio continuo de números reales de forma biunívoca [84]. Los vectores de palabras tiene la característica que capturan «el significado» y el contexto de las palabras en los valores asignados en las coordenadas de los vectores. Es decir, cualquier vector de reales es un posible embedding para una palabra dada, pero solo aquellos que capturen el significado y las relaciones con las otras palabras son considerados como válidos [55, 84, 7].

Esta representación es mejor que la anterior ya que sus vectores capturan analogías entre las palabras. Es decir, si x_{key} vector de palabra de “key”, el siguiente problema:

$$x_{\text{rey}} - x_{\text{hombre}} \approx x - x_{\text{mujer}}$$

Puede ser resuelto utilizando la similitud coseno mediante la siguiente ecuación,

$$x = \underset{\text{word} \in V}{\operatorname{argmax}} \operatorname{similarity}(\text{word}, x_{\text{rey}} - x_{\text{hombre}} + x_{\text{mujer}}) = x_{\text{reina}}$$

Donde la similitud coseno es una función entre $[-1, 1]$ que retorna uno si el ángulo comprendido entre los vectores es cero, es decir, si ambos vectores apuntan a un mismo lugar:

$$\operatorname{similarity}(x, y) = \cos(\theta) = \frac{x \cdot y}{\|x\|_2 \|y\|_2} = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

Utilizamos la similitud coseno y no la distancia euclidiana porque depende del ángulo entre los vectores de palabra y no de la magnitud de los mismos, esto hace que pueda capturar la relación entre «rey» y «hombre» o «mujer» y «reina» incluso si uno de ellos ocurre mucho en un corpus que el otro. La razón es porque la distancia euclidiana está influenciada por la frecuencia de aparición de las palabras en el corpus y la medida coseno es más resistente a las variaciones de ocurrencias entre términos que son semánticamente similares pero uno de ellos ocurre poco frecuente. A continuación presentamos una descripción de los modelos utilizados para construir vectores de palabras dado un corpus que contiene palabras en un lenguaje dado.

Word2Vec

Es un modelo de palabras predictivo ya que proporciona un vector de probabilidades a las palabras mediante el uso de redes neuronales artificiales. Estos métodos demostraron ser altamente efectivos para tareas como la analogía y la similitud entre palabras [55].

Word2vec no es solamente un algoritmo, sino una definición de dos técnicas: CBOV y el modelo Skip-gram. A continuación vamos a ilustrar cómo funcionan ambas:

- **Skip-gram.**

En este modelo la entrada es la representación mediante one-hot encoding de la palabra en un vocabulario V y la salida es el contexto de la palabra también representado por one-hot encoding, es decir, el conjunto de las C palabras alrededor de la palabra objetivo, representados por one-hot encoding. Entonces, el contexto de largo $C = 2$ para $\text{text} = \text{"fox"}$ en,

The quick brown fox jumps over the lazy dog

son las siguientes tuplas $\text{context_words} = \{(quick, brown); (brown, jumps); (jumps, over)\}$ es decir, aquellas palabras alrededor del objetivo.

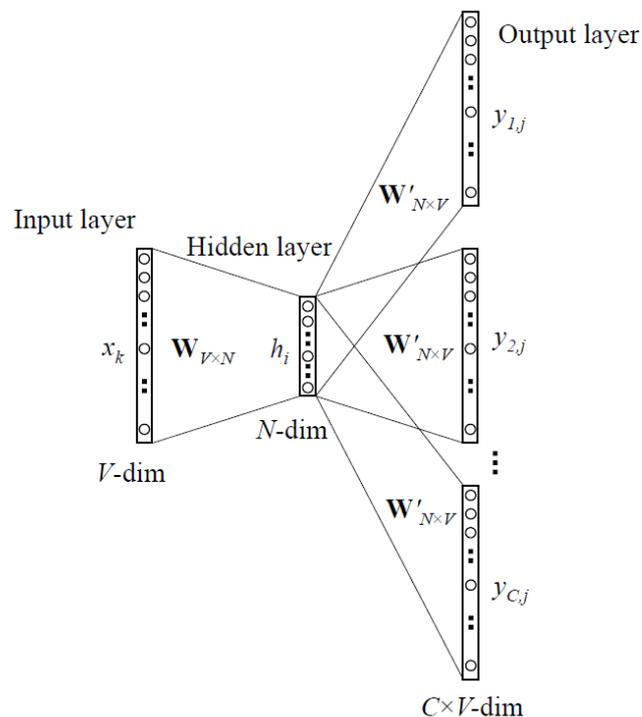


FIGURA 2.6: Vectores de palabras mediante Skip-ngram

Luego, como lo muestra la figura 2.6 se entrena una red neuronal de una sola capa oculta de tamaño N , de entrada one-hot de text y de C capas de salidas alimentadas por la representación en one-hot de context_words en tiempo de entrenamiento.

Finalmente, la representación por vectores de palabras para text se obtiene de los pesos de las N neuronas de la capa oculta.

- **CBOW** (Continuous bag of words).
 Las bolsas de palabras continuas son un método muy similar al anterior con excepción de que intercambia el espacio de entrada por el de salida. La idea principal detrás de este método es predecir dado un contexto cuál es la palabra que más se ajustaría.

Concretamente, como se puede ver en 2.7 la entrada del sistema es `context_word` de tamaño C codificado mediante one-hot y la salida es `text` representado de la misma manera. Luego se toma los pesos de la neurona de la capa oculta como representación de la palabra.

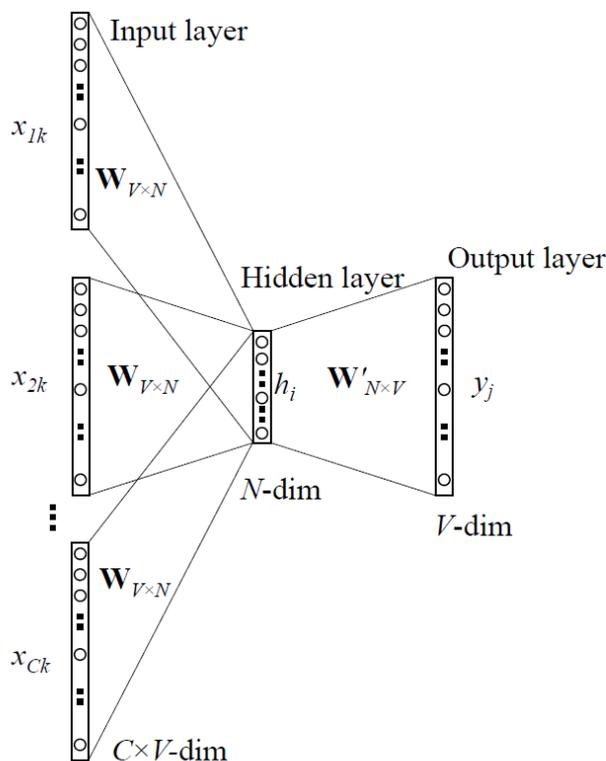


FIGURA 2.7: Vectores de palabras mediante CBOW

La diferencia sustancial entre ambos métodos es la forma con la que la red está entrenado. CBOW está diseñado para maximizar la probabilidad de encontrar la palabra objetivo mirando la lista de palabras en el contexto sin importar el orden en que las mire; es decir, las palabras más comunes van a tener más probabilidad de aparecer en los contextos que aquellas que no lo son. Mientras que Skip-ngram está implementado para deducir el contexto dado una palabra, es decir, palabras similares tendrán contextos similares.

Glove

Tanto CBOW como Skip-Grams son modelos predictivo que solo tienen en cuenta los contextos locales. Word2Vec no aprovecha el contexto global, es así que este método cobra sentido. Glove (Global Vectors) [30] utiliza la regresión logarítmica en combinación con la matriz de co-ocurrencia¹ y un algoritmo de factorización de matrices para producir vectores de palabras de contexto global.

Fasttext

Fasttext es otro método para representar palabras mediante vectores que utiliza la misma idea que word2vec donde agrega features de n-gram de caracteres. Entonces por ejemplo, para $n = 2$ la representación de fox en fasttext es $\{ef, fo, ox, xe\}$ donde las llaves indican el comienzo y el fin de la palabra. Una vez que la palabra ha sido representada usando n-gramas, se entrena un modelo "skip-ngram" para aprender los vectores de palabras.

2.5.3. Redes Neuronales Convolucionales

Una Red Neuronal Convolutiva (Convolutional Neural Network, CNN) [39] es un tipo especializado de red neuronal artificial utilizado para procesar datos en una topología de cuadrícula definida. En este tipo de red su aplicación es realizada en matrices n-dimensionales ya que permite extraer características en datos que requieran su representación en más de una dimensión (por ejemplo, imágenes) [29].

La operación de convolución

La convolución es una operación matemática entre dos funciones para producir una tercera función vista como el suavizado de una sobre la otra (i.e la convolución entre funciones es continuamente diferenciable tantas veces como lo son las dos funciones en total) [29]. Esta característica es la que le permite a la red neuronal aplicar filtros de suavizado y nitidez sobre las imágenes para poder extraer características para la clasificación.

Supongamos que representamos una imagen mediante una matriz I binaria donde la unidad indica la presencia de color, es decir

$$I : \{1, \dots, n_1\} \times \{1, \dots, n_2\} \rightarrow Y \subset \mathbb{R}, (i, j) \rightarrow I_{i,j}$$

Dado una matriz $K \in \mathbb{R}^{2h_1+1 \times 2h_2+1}$, la convolución discreta entre la imagen I y el filtro K viene dada por la siguiente ecuación [29]:

$$(I * K)_{i,j} \stackrel{\text{def}}{=} \sum_{u=-h_1}^{h_1} \sum_{v=-h_2}^{h_2} K_{u,v} I_{i+u,j+v}$$

Básicamente lo que sucede es que el filtro K se mueve sobre la superficie de la imagen I aplicando el operador de convolución y generando un nuevo elemento en $I * K$, la imagen suavizada de I [29]. Ejemplificando supongamos $n_1 = n_2 = 7$ y $K \in \mathbb{R}^{3 \times 3}$, la siguiente imagen ilustra este proceso

¹La matriz de co-ocurrencia es un arreglo bidimensional de las palabras en el vocabulario sobre sí misma donde cada elemento c_{ij} contiene el número de veces que la palabra i -ésima del vocabulario aparece en conjunto con la palabra j -ésima.

$$\underbrace{\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}}_I * \underbrace{\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}}_K = \underbrace{\begin{pmatrix} 1 & 4 & 3 & 4 & 1 \\ 1 & 2 & 4 & 3 & 3 \\ 1 & 2 & 3 & 4 & 1 \\ 1 & 3 & 3 & 1 & 1 \\ 3 & 3 & 1 & 1 & 0 \end{pmatrix}}_{I*K}$$

Donde en la figura se ve la creación del elemento $(I * K)_{1,4}$ mediante:

$$\begin{aligned}
 (I * K)_{1,4} &\stackrel{\text{def}}{=} \sum_{u=0}^2 \sum_{v=0}^2 K_{u,v} I_{1+u,4+v} = K_{0,0} I_{1,4} + K_{0,1} I_{1,5} + K_{0,2} I_{1,6} + \\
 &K_{1,0} I_{2,4} + K_{1,1} I_{2,5} + K_{1,2} I_{2,6} + \\
 &K_{2,0} I_{3,4} + K_{2,1} I_{3,5} + K_{2,2} I_{3,6} = 4
 \end{aligned}$$

En la red neuronal convolucional, K está incluido en una serie de filtros del núcleo (**Kernel Filters**). Obsérvese que para hacer más sencilla la suma los índices de K comenzaron desde 0. Luego se aplica a $(I * K)$ una función de activación no lineal punto a punto, para poder representar características más generales de la imagen, el propósito de esta operación es introducir la no linealidad en un sistema que básicamente ha estado computando operaciones lineales [29]. El resultado se denomina mapa de características (**feature maps**).

Hiperparámetros para efectuar la convolución

Hemos explicado la conectividad de cada neurona en la capa de convolución, sin embargo, no hemos discutido cuántas neuronas hay en el volumen de salida o cómo están organizadas. Para este propósito existen tres hiperparámetros que controlan la dimensión de salida [29].

- **Profundidad (Depth)**. Se corresponde con el número de filtros utilizados en la convolución, y cada uno aprende a buscar una característica diferente en la entrada. Formalmente, en la convolución aplicamos un conjunto $\{K_1, \dots, K_{\text{depth}}\}$ de filtros, donde cada uno puede activarse según propiedades en la entrada (manchas de color en una imagen, bordes, etc..)
- **Avance del filtro (Stride)**. Se refiere al paso con el cual el filtro se va a deslizar. Si consideramos I como una representación de una imagen por píxeles, cuando $\text{stride} = 1$ el filtro de mueve un píxel a la vez, cuando $\text{stride} = 2$ los filtros saltan dos píxeles a la vez a medida que deslizamos el filtro, y así sucesivamente. La figura 2.8 ilustra este proceso.
- **Relleno de ceros (Zero-Padding)** A veces es conveniente rellenar el volumen de entrada con ceros alrededor del borde. Esta característica es la que nos permite preservar el tamaño de entrada.

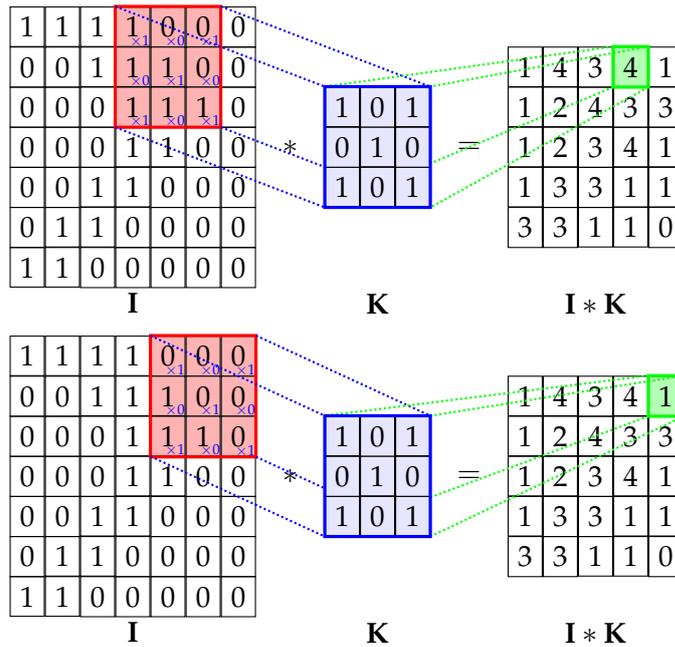


FIGURA 2.8: Avance del filtro $\text{stride} = 1$

Así es que podemos calcular la dimensión de la matriz de convolución de salida mediante la de entrada, el filtro y el avance en el que se aplican los mismos, y la cantidad de ceros rellenados en el borde [29]. Entonces, en dos dimensiones, si $I \in \mathbb{R}^{n_1 \times n_2}$, $K \in \mathbb{R}^{h_1 \times h_2}$, P es el número de ceros aplicados al borde y S es el avance del filtro, la dimensión de salida viene dado por

$$\frac{(n_1, n_2) - (h_1, h_2) + 2\vec{P}}{S} + \vec{1}$$

Luego, en el ejemplo de la figura 2.8 tenemos que $n_1 = n_2 = 7$, $h_1 = h_2 = 3$, no utilizamos ningún relleno así que $\vec{P} = \vec{0}$ y el filtro se mueve una vez por píxel, por ende $S = 1$, luego la dimensión de la matriz de salida es,

$$\frac{(n_1, n_2) - (h_1, h_2) + 2\vec{P}}{S} + \vec{1} \Leftrightarrow \frac{(7, 7) - (3, 3) + 2 \cdot \vec{0}}{1} + (1, 1) \Leftrightarrow \frac{(4, 4)}{1} + (1, 1) = (5, 5) \Leftrightarrow (I * K) \in \mathbb{R}^{5 \times 5}, \text{ como se puede ver en 2.8}$$

Obsérvese que si usamos un relleno de ceros por el borde $P = 1$, tenemos que $(I * K) \in \mathbb{R}^{7 \times 7}$ preservando el tamaño de entrada de la red [29].

2.6. Trabajos relacionados.

El surgimiento de competencias como SemEval¹ y TASS², dedicados a promover el desarrollo de técnicas para el procesamiento de lenguaje natural, han contribuido en la generación de actividades para el diseño y desarrollo de sistemas de ABSA. En estas competencias se parte de un corpus anotado con las quintuplas 2.2.1 por texto en el corpus y los participantes buscan soluciones a ciertas tareas dentro de este marco. Por ejemplo en SemEval 2016 [46], los participantes utilizan un corpus anotado con la siguiente estructura:

```

1 <text>
2   Buen servicio , ambiente acogedor y tranquilo .
3 </text>
4 <Opinions>
5   <Opinion target="servicio "
6         category="SERVICE#GENERAL"
7         polarity="positive "
8         from="6 "
9         to="13" />
10  <Opinion target="ambiente "
11        category="AMBIENCE#GENERAL"
12        polarity="positive "
13        from="16 "
14        to="23" />
15 </Opinions>

```

Y desarrollan soluciones a las siguientes subtareas de este marco:

- **Nivel de oración (Sentence-Level).** Dado una oración que involucra una opinión sobre una entidad, realizar las siguientes actividades,
 - SB1. **Categoría del aspecto.** Dado una entidad “E”, recuperar el aspecto “A” sobre un conjunto de valores predefinidos y etiquetar el texto con “category=E#A”.
 - SB2. **Recuperación de la expresión del objetivo de opinión.** Extracción de la expresión lingüística utilizada en el texto dado para referirse a la entidad “E” de cada par E#A. La expresión se define por sus desplazamientos iniciales y finales. Cuando no hay una mención explícita su valor es “null”.
 - SB3. **Cálculo de la polaridad.** Asignar a cada par E#A identificado una de las siguientes etiquetas de polaridad: “positive”, “negative”, “neutral” (levemente positivo o levemente negativo).
- **Nivel de texto (Text-Level).** Dado una reseña de un cliente sobre un dominio predefinido, identificar las tuplas definidas mediante (E#A,p) que resumen las opiniones expresadas en la revisión. Donde p se puede configurar como “positive”, “negative”, “neutral” o “conflict”. Entonces por ejemplo, el texto de la reseña «Buen servicio, ambiente acogedor y tranquilo» queda configurado como (SERVICE#GENERAL,positive), (AMBIENCE#GENERAL,positive)

¹https://aclweb.org/aclwiki/SemEval_Portal

²<http://www.sepln.org/workshops/tass/>

- **ABSA fuera del dominio.** Los participantes de SemEval 2016 [46] tuvieron la oportunidad de probar sus sistemas en dominios para los que no se proporcionaron datos de entrenamiento; estos corpus permanecieron desconocidos hasta el inicio del período de evaluación.

La mayoría de los sistemas de aprendizaje en SemEval-2014 [48] o SemEval-2015 [48] dedicados a ABSA utilizaban los algoritmos de aprendizaje tradicionales como SVM [37, 10, 11, 88] o CRF [83, 18, 82, 25] con ingeniería de atributos de forma manual para poder realizar el aprendizaje. Sin embargo, en aquel momento las redes neuronales profundas, particularmente las CNN [39], ya se estaban desempeñando mucho mejor que los métodos tradicionales [94, 3, 22] que luego estas ideas fueron utilizadas en SemEval-2016 [46], mas aun el grupo que obtuvo los mejores resultados en la tarea de ABSA fue NLANGP [96] quienes realizaron un modelo secuencial de etiquetado utilizando redes neuronales profundas. Finalmente queremos observar que la tareas en TASS son muy similares a las de SemEval, con la salvedad que los corpus proporcionados tienen distinta estructura y formato.

En este proyecto vamos a construir un sistema de aprendizaje que tome los corpus de estas competencias, los unifique en un único formato y realice las actividades mencionadas anteriormente a excepción de SB2., debido a la insuficiencia de datos de entrenamiento.

Capítulo 3

Descripción general de la solución

En este capítulo realizamos una visión general de la solución propuesta al problema de recuperación automática de las opiniones de los usuarios contenidas en los textos.

3.1. Arquitectura de la solución

Recapitulando, el objetivo es obtener de forma automática las siguientes ternas de los textos de los documentos:

(entity, aspect, polarity)

Nuestra propuesta se basa en construir un sistema de aprendizaje sobre los textos de los corpus para las competencias TASS y SemEval de manera que identifique las entidades, sus aspectos y que calcule la valoración (positiva, negativa, neutral) de la orientación del sentimiento basado en los aspectos. Esta propuesta es análoga a completar las subtarefas definidas en 2.6, a excepción de SB2., que escapa del alcance de este proyecto.

La figura 3.1 se corresponde con la visión general de la arquitectura definida para este propósito, la misma puede ser descrita mediante las siguientes seis etapas:

- Etapas 1.** Dado los conjuntos de entrenamiento de TASS o SemEval, producir un nuevo corpus con una estructura común, que luego sea consumida por un sistema de aprendizaje.
- Etapas 2.** Preprocesar los textos dentro del corpus de entrenamiento producido en la etapa anterior.
- Etapas 3.** Determinar el dominio del sistema en base a la estructura 3.2, utilizando los datos de entrenamiento de la primer etapa.
- Etapas 4.** Dado un conjunto de datos estructurados de la primer etapa, determinar los atributos de salida del sistema.
- Etapas 5.** Entrenar utilizando las salidas producidas por los anteriores puntos.
- Etapas 6.** Dado el conjunto de validación y las salidas del clasificador del punto anterior, realizar la evaluación del algoritmo de aprendizaje e imprimir las métricas asociadas al mismo.

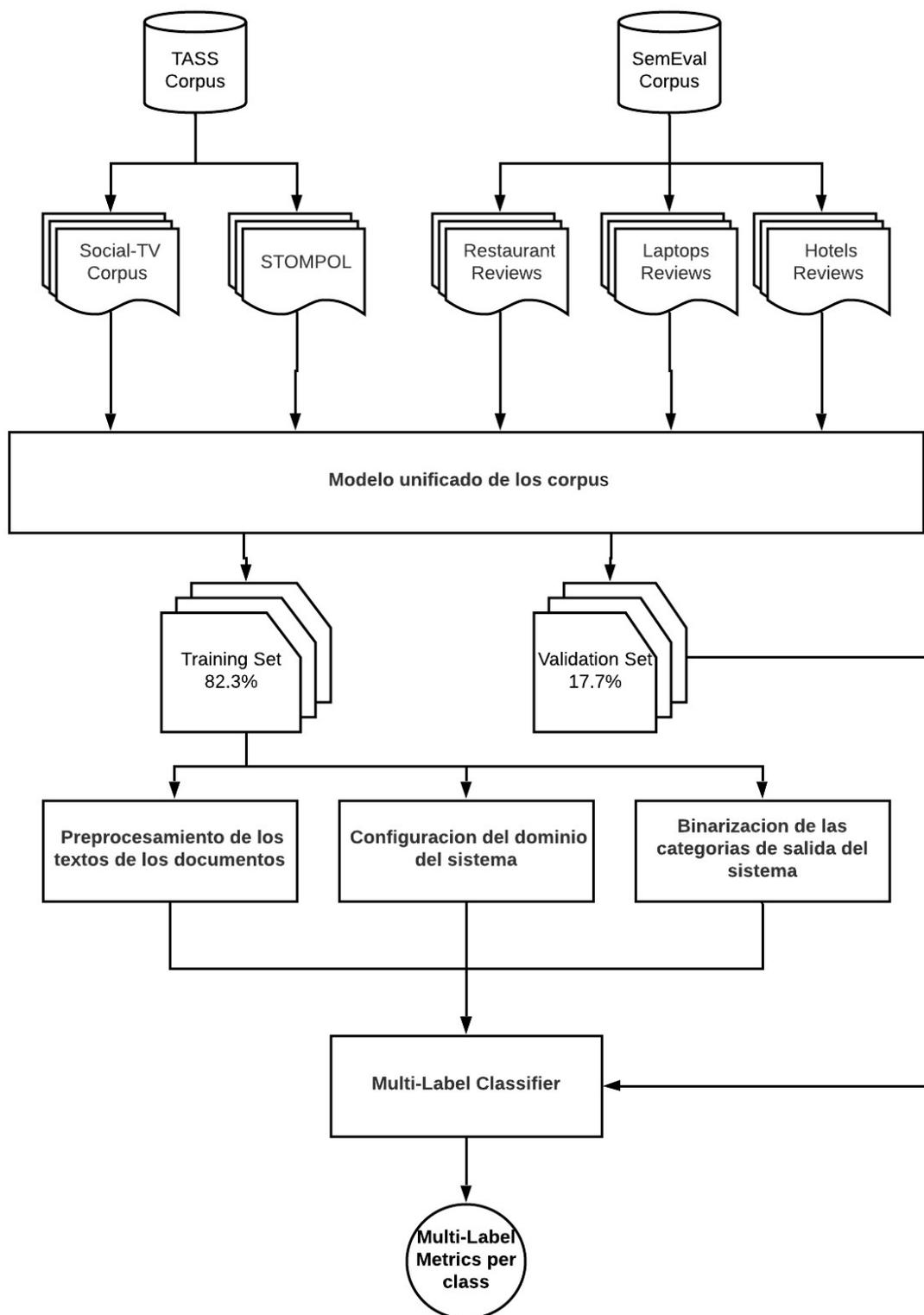


FIGURA 3.1: Arquitectura de la solución

Las próximas secciones realizan un análisis más específico de los componentes de la arquitectura desarrollando por separado las distintas etapas mencionadas anteriormente.

El primer gran bloque de la arquitectura procesa los corpus y los transforma a una estructura común, donde luego va a ser consumida por un algoritmo de aprendizaje que necesita que los documentos tengan un formato específico para poder determinar todos los parámetros para la clasificación. La salida de este módulo es un nuevo corpus con la información en el formato correcto. El algoritmo termina con la separación en datos de entrenamiento y en datos de validación.

Los siguientes tres módulos importantes realizan actividades sobre el conjunto de datos de entrenamiento y validación. El módulo de procesamiento, recupera los textos de los documentos, elimina su ruido y luego los representa mediante una lista de vectores de palabras. El sistema de configuración, recupera del corpus el dominio del sistema 3.2 y lo almacena en una base de datos para que luego el algoritmo lo utilice para configurar las variables del sistema. El tercero, es el componente de binarización que determina las clases de salida del algoritmo.

En último lugar, el sistema utiliza la salida de los tres componentes para realizar el análisis basado en un algoritmo de clasificación multietiqueta (Multi-Label Classification Algorithm) que se basa en asignar un conjunto de etiquetas a un ejemplo de entrenamiento. El mismo contiene la información necesaria de los pasos anteriores para que, luego de realizar la clasificación, transforme el conjunto de evaluación para que quede preparado para el cálculo de métricas.

Finalmente, la arquitectura culmina con la evaluación de resultados. El capítulo 4 de este documento, es el encargado de presentar los detalles del funcionamiento del algoritmo y el capítulo 5 el encargado de exponer los métodos de evaluación y resultados.

3.2. Modelo de dominio

Esta sección es la encargada describir la estructura del corpus de opiniones definido para la clasificación. Como habíamos adelantado, la idea general es definir una nueva estructura que sea utilizada por el algoritmo de aprendizaje y traducir los corpus de TASS y SemEval a esta nueva estructura para realizar la clasificación de forma independiente del formato del corpus que estemos tratando.

La figura 3.2 ilustra la estructura del corpus por la cual se va a realizar el tratamiento de las opiniones del problema.

```

1 <Domain name=" ... "
2   categories=" ... "
3   entities=" ... "
4   aspects=" ... "
5   relation_table=" ... ">
6   <opinions>
7     <opinion>
8       <text>...</text>
9       <entities>
10        <entity aspect=" ... "
11          aspect-cue=" ... "
12          category=" ... "
13          entity=" ... "
14          entity-cue=" ... "
15          polarity=" ... "
16          polarity-cue=" ... " />
17        ...
18      </entities>
19    </opinion>
20    ...
21  </opinions>
22 </Domain>

```

FIGURA 3.2: Estructura del corpus de opiniones

Definimos una estructura por la cual se asocia un dominio a una colección de opiniones donde cada una contiene un conjunto de entidades y sus aspectos heredados por la categoría de la entidad.

Obsérvese que utilizamos el atributo "relation_table" para almacenar los posibles aspectos que una entidad puede tener basándose en la categoría que la contiene. Gracias a que estamos trabajando una jerarquía de dos niveles, podemos representar estas relaciones como una tabla y almacenarlo en formato de texto plano en el corpus, para que luego pueda ser utilizado por el sistema de aprendizaje.

A continuación describimos las etiquetas asociadas a la estructura del corpus:

- **Domain**: Define el dominio del sistema que se va a estudiar.
 - **name**: Nombre del dominio utilizado para identificar el corpus.
 - **categories**: Representa la lista de categorías del sistema. La misma es representada como una secuencia de palabras separadas por espacio, es decir “category_1 category_2 ... category_n”.
 - **entities**: Lista de entidades del sistema. Representada de la misma forma que la lista de categorías “entities_1 entities_2 ... entities_n”.
 - **aspects**: Lista de aspectos del sistema. Representado de la misma forma que los dos anteriores atributos, “aspects_1 aspects_2 ... aspects_n”.
 - **relation_table**: Tabla que lista los aspectos de las categorías del sistema. Se representa mediante una tabla en formato JSON.
- **opinions**: Define el conjunto de opiniones del sistema.
- **opinion**: Define una opinión en particular, compuesta por el texto de la opinión y un conjunto de entidades con sus aspectos y la polaridad asociada a los mismos.
- **text**: Contiene el texto de la opinión.
- **entities**: Define el conjunto de entidades asociadas a la opinión.
- **entity**: Define una entidad en particular.
 - **aspect**: Es el aspecto asociado a la entidad de acuerdo con la definición 2.2.3.
 - **aspect-cue**: Contiene la expresión de aspecto en el texto.
 - **category**: Contiene la categoría asociada a la entidad.
 - **entity**: Es la entidad asociada a la opinión 2.2.2.
 - **entity-cue**: Contiene la presencia de la entidad en el texto.
 - **polarity**: Es la orientación del sentimiento del aspecto asociado a la entidad 2.2.1.
 - **polarity-cue**: Etiqueta asociada a la presencia de la polaridad en el texto. La misma es una palabra o secuencia de palabras en el texto que indica la presencia de una valoración por parte del autor de la opinión.

3.2.1. Traducción de corpus utilizados al modelo definido

Con esta estructura se puede traducir corpus utilizados en las competencias SemEval 2.4.1 y TASS 2.4.2.

Traducción sobre Restaurant Reviews

Utilizando las definiciones en 2.4.1 sobre el corpus de SemEval, podemos definir el dominio del sistema para configurar los parámetros necesarios para luego desarrollar un algoritmo que realice la traducción de las reseñas al corpus de opiniones.

De acuerdo al esquema 3.2, estructuramos el dominio del corpus de restaurantes de la siguiente manera:

Domain:

- name = "Restaurant Review"
- categories = FOOD, DRINKS, SERVICE, AMBIENCE, LOCATION, RESTAURANT
- entities = ALL
- aspects = GENERAL, PRICES, QUALITY, STYLE&OPTIONS, MISCELLANEOUS
- relation_table definida mediante la siguiente tabla,

Categories Aspects	GENERAL	PRICES	QUALITY	STYLE&OPTIONS	MISCELLANEOUS
RESTAURANT	✓	✓	✗	✗	✓
FOOD	✗	✓	✓	✓	✗
DRINKS	✗	✓	✓	✓	✗
AMBIENCE	✓	✗	✗	✗	✗
SERVICE	✓	✗	✗	✗	✗
LOCATION	✓	✗	✗	✗	✗

FIGURA 3.3: Conjunto de relaciones para restaurant review

Observaciones

- Esto es posible ya que tenemos una representación de las entidades en dos niveles utilizando el término aspecto 2.2.3 para denotar los componentes y atributos de una entidad, es decir podemos representar $R_{\text{restaurant}}$ mediante una tabla que asocia a cada categoría la lista de aspectos que le corresponde.
- En este corpus, los autores decidieron no utilizar entidades pero sí hablar sobre categorías de entidades. Denotamos a la entidad "ALL" en nuestro dominio, aquella entidad que se refiere a la categoría como un todo.

El siguiente documento 3.4 se corresponde con un fragmento del corpus *Restaurant Reviews* definido por la SemEval.

```
1 <Reviews>
2   <Review rid="1004293">
3     <sentences>
4       <sentence id="1032695:1">
5         <text>
6           Everything is always cooked to perfection,
7           the service is excellent,
8           the decor cool and understated.
9         </text>
10        <Opinions>
11          <Opinion target="service" category="SERVICE#GENERAL"
12                polarity="positive" from="47" to="54"/>
13          <Opinion target="decor" category="AMBIENCE#GENERAL"
14                polarity="positive" from="73" to="78"/>
15        </Opinions>
16      </sentence>
17    </sentences>
18  </Review>
19 </Reviews>
```

FIGURA 3.4: Fragmento del corpus Restaurant Reviews 2.4.1

En 3.4 la entidad RESTAURANT representa al restaurante de la opinión como un todo. Nosotros creemos que en la teoría, RESTAURANT debería ser asociado a una categoría de entidades más que a una entidad en sí, porque luego lo ideal sería realizar una comparativa de restaurantes utilizando entidades.

Ejemplificando, supongamos que nuestra ciudad tiene dos restaurantes llamados "Resto1" y "Resto2". En el corpus de reseñas de restaurantes propuesto por la SemEval, la entidad a analizar es $e = \text{RESTAURANT}$ y la misma no permite distinguir entre ambos. Sin embargo en nuestra propuesta vamos a tener una categoría RESTAURANTES que incluye a dos entidades $e_1 = \text{RESTO1}$ y $e_2 = \text{RESTO2}$.

Exactamente el mismo problema sucede con la definición de las entidades en juego de este corpus. Su definición es tan general que no permite el estudio particular de entidades en el dominio.

A raíz de estas observaciones realizamos la traducción a nuestro modelo, la figura 3.5, se corresponde con la salida de nuestro algoritmo:

```

1 <Domain name="Restaurant Review"
2   categories="FOOD DRINKS SERVICE
3     AMBIENCE LOCATION RESTAURANT"
4   entities="ALL"
5   aspects="GENERAL LOVE PRICES
6     QUALITY STYLE&OPTIONS
7     MISCELLANEOUS"
8   relation_table="...">
9 <opinions>
10 <opinion>
11 <text>
12   Everything is always cooked to perfection,
13   the service is excellent,
14   the decor cool and understated.
15 </text>
16 <entities>
17   <entity aspect="GENERAL"
18     aspect-cue="NULL"
19     category="SERVICE"
20     entity="ALL"
21     entity-cue="service"
22     polarity="positive"
23     polarity-cue="NULL"/>
24   <entity aspect="GENERAL"
25     aspect-cue="NULL"
26     category="AMBIENCE"
27     entity="ALL"
28     entity-cue="decor"
29     polarity="positive"
30     polarity-cue="NULL"/>
31 </opinion>
32 </opinions>
33 </Domain>

```

FIGURA 3.5: Traducción sobre corpus 2.4.1

Donde se no se muestra el contenido de `relation_table` a efectos de que la figura entre en la página. Notar que, mientras que 3.4 la categoría denota al par (e, a) , representado por $e#a$, donde e es la entidad y a el aspecto en juego; en la traducción 3.5 distinguimos la categoría de la entidad y representamos a la mismo como un todo, es decir, $e = ALL$ incluida en la categoría RESTAURANT refiere a la entidad que aparece en una opinión refiriéndose a restaurantes como un todo y no a un restaurante particular.

Traducción sobre Social-TV Corpus

Las definiciones de 2.4.2 sobre el corpus SocialTV del TASS, nos permite definir el dominio mediante la siguiente estructura:

Domain:

- name = "Social-TV"
- categories = Afición, Árbitro, Autoridades, Entrenador, Equipos, Jugadores, Partido, Retransmisión
- entities = Atlético_de_Madrid, Barcelona, Real_Madrid, Otro_Equipo, Alexis_Sánchez, Alvaro_Arbeloa, Andrés_Iniesta, Angel_Di_María, Asier_Ilarramendi, Carles_Puyol, Cesc_Fábregas, Cristiano_Ronaldo, Dani_Alves, Dani_Carvajal, Fábio_Coentrão, Gareth_Bale, Iker_Casillas, Isco, Javier_Mascherano, Jesús_Rodríguez, José_Manuel_Pinto, Karim_Benzema, Lionel_Messi, Luka_Modric, Marc_Bartra, Neymar_Jr, Pedro_Rodríguez, Pepe Sergio_Busquets, Sergio_Ramos, Xabi_Alonso, Xavi_Hernández, Otro_Jugador
- aspects = GENERAL
- relation_table definida mediante la siguiente tabla,

Categories Aspects	GENERAL
Afición	✓
Árbitro	✓
Autoridades	✓
Entrenador	✓
Equipos	✓
Jugadores	✓
Partido	✓
Retransmisión	✓

Obsérvese que existen categorías que no tienen entidades asociadas como lo es por ejemplo "Arbitro". Pero además en este modelo se está evaluando a todas las entidades como un todo sin tener en cuenta los aspectos de las mismas, razón por la cual solo existe un único aspecto, el cual es, por defecto, el aspecto GENERAL.

Traducción sobre STOMPOL

La descripción del corpus STOMPOL en 2.4.2 del TASS, nos permite construir el dominio del sistema mediante la siguiente estructura:

Domain:

- name = "STOMPOL"
- categories = Partidos_Políticos_Españoles
- entities = Partido_Popular, Partido_Socialista_Obrero_Español, Izquierda_Unida, Podemos, Ciudadanos, Unión_Progreso_y_Democracia
- aspects = GENERAL, Economía, Sanidad, Educación, Propio_partido, Otros_aspectos
- relation_table definida mediante la siguiente tabla,

Categories Aspects	GENERAL	Economía	Sanidad	Educación	Propio_partido	Otros_aspectos
Partidos_Políticos_Españoles	✓	✓	✓	✓	✓	✓

Obsérvese que STOMPOL es un corpus sobre comentarios de los distintos partidos políticos españoles. Esta categoría no aparece en 2.4.2 pero fue necesaria su definición para poder describir el dominio del sistema completamente.

A continuación 3.6 un fragmento del corpus traducido:

```

1 <Domain name="STOMPOL"
2   categories="Partidos_Políticos_Españoles"
3   entities="Partido_Popular Partido_Socialista_Obrero_Español
4             Izquierda_Unida Podemos Ciudadanos
5             Unión_Progreso_y_Democracia"
6   aspects="Economía Sanidad Educación
7             Propio_partido Otros_aspectos"
8   relation_table="... ">
9 <opinions>
10  <opinion>
11    <text>
12      @ahorapodemos @Pablo_Iglesias_ @SextaNocheTV
13      Que alguien pregunte si habrá cambios en las becas
14      MEC para universitarios , por favor.
15    </text>
16    <entities>
17      <entity aspect="Educacion"
18              aspect-cue=" universitarios "
19              category="Partidos_Políticos_Españoles "
20              entity="Podemos"
21              entity-cue="@ahorapodemos"
22              polarity="NEU"
23              polarity-cue="NULL" />
24    </entities>
25  </opinion>
26 </opinions>
27 </Domain>

```

FIGURA 3.6: Fragmento del corpus STOMPOL formateado 2.4.2

Donde podemos notar que la categoría de cada entidad queda predefinida ya que todas las entidades se agrupan en una misma: «Partidos Políticos Españoles en cierto periodo».

3.3. Preprocesamiento de los textos de los documentos

Dado que los textos de las opiniones son la unidad de información escrita por los autores de opinión, los mismos varían en estructura y forma, por lo que fue necesario unificarlos a una estructura en común. Este tipo de actividades en el cual «limpiamos» los textos de los documentos acomodándolos a una estructura procesable por un algoritmo de aprendizaje se llama *Preprocesamiento*.

Las tres propiedades deseables en un corpus de opiniones son [57]:

1. **Representativo de todo el espacio.** Es decir, que contenga ejemplos de entrenamientos que representen lo mejor posible la realidad planteada.
2. **Ausencia de ruido.** El conjunto de datos no está incompleto, corrupto ni muestra inconsistencias o reitera información.
3. **Tamaño suficientemente grande.** Necesitamos tener los ejemplos de entrenamiento suficientes que le permita al algoritmo poder generalizar los datos de forma efectiva.

En esta sección nos enfocaremos fuertemente en mejorar el segundo punto de manera tal de presentarle a nuestro algoritmo una estructura de datos de texto lo suficientemente sólida como para que el mismo pueda aprender de forma eficiente. A continuación, la serie de tareas realizadas sobre los mismos:

1. Eliminación de ruido

Es la tarea de unificar el texto a un formato común y remover toda secuencia de caracteres que no contenga información específica para la tarea que vamos a realizar. Las siguientes son actividades realizadas dentro de esta tarea,

- Transformar los textos a minúsculas.
- Remover direcciones de correo.
- Remover URLs.
- Remover comas y puntos.
- Sustitución de saltos de líneas a un espacio en blanco.
- Sustitución de espacios en blancos seguidos a un solo espacio en blanco.
- Eliminación de censuras (Algunos corpus, censuran las malas palabras mediante por ejemplo “#\$& %\$”).
- Eliminación de signos de pregunta y exclamación.
- Eliminación de letra repetidas. Por ejemplo, para el español «bieeen» se sustituye por «bien» y para el inglés «cool» por «cool».
- Sustitución de emoticones felices por «feliz» y de tristes por «triste».
- Eliminación de caracteres no alfanuméricos dentro de las palabras. Por ejemplo, «pre-procesamiento» por «preprocesamiento».

2. Eliminación de Stopwords

Es la tarea de eliminar una colección de palabras que no aporta significado alguno a la clasificación. Para este propósito utilizamos la librería NLTK ¹, la cual ya viene con lista de palabras predefinidas para ambos idiomas (Español e Inglés).

¹<https://www.nltk.org/>

3. Tokenización

Es el proceso por el cual convertimos nuestro texto a una lista de palabras claves denominadas “Tokens” permitiendo tener una estructura procesable por un algoritmo. También se utilizó la librería NLTK para esta actividad.

Luego de la lista de tareas realizadas anteriormente, necesitamos armar un vocabulario con las palabras que aparecen en los textos. Es decir, construir una función que retorne para cada token un número identificador comenzando desde cero de forma ascendente. Vamos a utilizar el ejemplo del corpus de Restaurantes ya formateado 3.5 para ilustrar la secuencia de pasos realizados hasta ahora. Los textos de las opiniones extraídos del ejemplo son los siguientes:

```
<text>Servicio impecable y se come de maravilla a buen precio!</text>
<text>Nos pusieron en la terraza, sin avisar previamente.</text>
```

Es así que entonces tendremos como salida las siguientes listas de palabras,

```
[servicio, impecable, come, maravilla, buen, precio]
[pusieron, terraza, avisar, previamente]
```

Realizamos una unión de todas las palabras en el corpus (sin repetir) y construimos una función *vocabulary* que asigna un identificador a cada una de ellas de la siguiente manera:

$$vocabulary(word) = \begin{cases} 1 & \text{si word=servicio} \\ 2 & \text{si word=impecable} \\ \vdots & \vdots \\ 8 & \text{si word=terrazza} \\ 9 & \text{si word=avisar} \\ 10 & \text{si word=previamente} \end{cases} \quad (3.1)$$

El vocabulario es una función dentro de nuestro sistema que nos retorna el identificador de la palabra en el vocabulario. La idea es, utilizar esta función para representar las palabras del vocabulario mediante vectores de palabras como se explicó en 2.5.2.

Entonces, si llamamos I a los índices asociados por la función de vocabulario, construimos una función *embedding* que a cada palabra $word \in W = Dom(vocabulary)$ le asigna un vector en el espacio n -dimensional, es decir:

$$embedding : I \longrightarrow \mathbb{R}^n, \text{ donde } n \in \mathbb{N}$$

$$embedding(index) = (x_1, \dots, x_n) \quad x_i \in \mathbb{R} \quad \forall i \in \{1, \dots, n\}$$

$$vocabulary(word) = index \text{ donde, word es la palabra que se corresponde al vector.}$$

Este mapeo es necesario para poder representar los documentos de textos a un formato que luego pueda ser consumido por un algoritmo de aprendizaje, ya que los mismos sólo distinguen entre números para poder aprender.

Nosotros utilizamos un conjunto de vectores de palabras pre-entrenados mediante los métodos vistos en 2.5.2. La sección 4.1 realiza una lista de los vectores

de palabras utilizados así como sus respectivas referencias.

Finalmente, definimos la matriz de vectores de palabras como una lista de vectores donde cada fila i se corresponde con el vector de palabra del i -ésimo índice en el vocabulario.

$$E = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1i} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2i} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ x_{(k-2)1} & x_{(k-2)2} & \dots & x_{(k-2)i} & \dots & x_{(k-2)n} \\ x_{(k-1)1} & x_{(k-1)2} & \dots & x_{(k-1)i} & \dots & x_{(k-1)n} \\ x_{k1} & x_{k2} & \dots & x_{ki} & \dots & x_{kn} \end{pmatrix} \begin{array}{l} \text{vocabulary}(\text{Servicio}) \leftarrow 1 \\ \text{vocabulary}(\text{impecable}) \leftarrow 2 \\ \vdots \\ \text{vocabulary}(\text{terrace}) \leftarrow k-2 \\ \text{vocabulary}(\text{avisar}) \leftarrow k-1 \\ \text{vocabulary}(\text{previamente}) \leftarrow k \end{array}$$

vectores de palabras de dimensión n por cada fila del vocabulario

La dimensión de la matriz es kn ; $k = |W|$ y n el tamaño de los vectores de palabras. Esta estructura es la utilizada para representar los textos dentro del algoritmo.

3.4. Representación de las clases de salida

En esta sección es la encargada de desarrollar las ideas detrás del componente de Binarización. Como lo habíamos adelantado en 3.1, su objetivo es el de representar las opiniones del corpus como etiquetas de salida de un algoritmo de aprendizaje.

Obsérvese que, como potencialmente un texto puede tener un cantidad arbitraria de opiniones, las etiquetas de salidas necesitan reflejar este comportamiento. Nos enfrentamos al problema de modelar las posibles opiniones de un texto como una serie de etiquetas predefinidas que permiten al algoritmo de aprendizaje inferir las ternas de un ejemplo no visto.

Vamos a utilizar el corpus de restaurantes 2.4.1 a modo de ilustrar las actividades realizadas por este componente. En base al conjunto de relaciones del corpus en `relation_table` tenemos:

Categories Aspects	GENERAL	PRICES	QUALITY	STYLE&OPTIONS	MISCELLANEOUS
RESTAURANT	✓	✓	✗	✗	✓
FOOD	✗	✓	✓	✓	✗
DRINKS	✗	✓	✓	✓	✗
AMBIENCE	✓	✗	✗	✗	✗
SERVICE	✓	✗	✗	✗	✗
LOCATION	✓	✗	✗	✗	✗

Además, el conjunto de orientaciones de sentimiento del corpus está definido mediante `polarities = {P, N, NEU}`.

Como el conjunto de entidades del dominio `entities = {ALL}`, las posibles combinaciones para los objetivos de opinión (Opinion target - OT, i.e. pares (entity, aspect)) que están en `relation_table` son las siguientes:

$$\text{opinion_targets} = \{(\text{ALL}, \text{GENERAL}); (\text{ALL}, \text{PRICES}); \quad (3.2)$$

$$(\text{ALL}, \text{QUALITY}); (\text{ALL}, \text{STYLE\&OPTIONS});$$

$$(\text{ALL}, \text{MISCELLANEOUS})\}$$

Utilicemos el siguiente fragmento del corpus de restaurantes para presentar los métodos desarrollados para crear etiquetas de salida en base al conjunto anterior:

```

1 <Domain name="Restaurant Review"
2   categories="FOOD DRINKS SERVICE
3     AMBIENCE LOCATION RESTAURANT"
4   entities="ALL"
5   aspects="GENERAL LOVE PRICES
6     QUALITY STYLE&OPTIONS
7     MISCELLANEOUS">
8 <opinions>
9   <opinion>
10    <text>
11      Servicio impecable y se come de maravilla a buen precio!
12    </text>
13    <entities>
14      <entity aspect="GENERAL" aspect-cue="NULL" category="SERVICE" entity="ALL"
15        entity-cue="NULL" polarity="positive" polarity-cue="NULL"/>
16      <entity aspect="GENERAL" aspect-cue="NULL" category="RESTAURANT" entity="ALL"
17        entity-cue="NULL" polarity="positive" polarity-cue="NULL"/>
18      <entity aspect="PRICES" aspect-cue="precio" category="RESTAURANT" entity="ALL"
19        entity-cue="NULL" polarity="positive" polarity-cue="buen"/>
20    </entities>
21  </opinion>
22  <opinion>
23    <text>
24      Nos pusieron en la terraza , sin avisar previamente.
25    </text>
26    <entities>
27      <entity aspect="GENERAL" aspect-cue="NULL" category="LOCATION" entity="ALL"
28        entity-cue="terracea" polarity="negative" polarity-cue="NULL"/>
29    </entities>
30  </opinion>
31 </opinions>
32 </Domain>

```

FIGURA 3.7: Ejemplo ilustrativo: Fragmento de Restaurant Review formateado 3.5

Luego, el módulo de unificación lo formatea y lo representa de la siguiente manera:

Textos	Categorías	Opiniones
text ₁	SERVICE	{(ALL, GENERAL, P)}
text ₁	RESTAURANT	{(ALL, GENERAL, P), (ALL, PRICES, P)}
text ₂	LOCATION	{(ALL, GENERAL, P)}

FIGURA 3.8: Corpus para ABSA: Restaurant Reviews

Donde los textos del corpus se representan por variables a modo de facilidad en la visualización de los corpus:

text₁ = Servicio impecable y se come de maravilla a buen precio!

text₂ = Nos pusieron en la terraza , sin avisar previamente.

Nuevamente, dado el par de entrada (text,category) necesitamos obtener el conjunto de opiniones asociadas, es decir, recuperar el objetivo de la opinión (e, a) y la polaridad.

Obsérvese que el texto de la segunda fila en el corpus 3.8 tiene un conjunto de dos opiniones, mostrando que la clase de salida "Opiniones", no es categórica, por lo que el problema no se puede resolver con algoritmos de aprendizaje multi-clase.

Es así, que el objetivo principal de este módulo es transformar la salida del corpus para que luego, pueda ser resuelta por un algoritmo de aprendizaje desarrollado en un siguiente módulo en la arquitectura. Para cumplir este propósito, desarrollamos dos métodos:

Método 1

El primer método consiste en utilizar las relaciones entre las entidades y los aspectos del sistema definido por `relation_table` y definir nuevas etiquetas construidas de la siguiente manera:

Si $(entity, aspect)$ es un posible objetivo de opinión en 3.2, construimos la etiqueta de salida "entity-aspect-polarity" donde *polarity* es una posible orientación de sentimiento en *polarities*.

Este método en el corpus de restaurantes 3.8 produce el siguiente conjunto de etiquetas,

```
{ ALL-GENERAL-P, ALL-GENERAL-N, ALL-GENERAL-NEU,
  ALL-PRICES-P, ALL-PRICES-N, ALL-PRICES-NEU,
  ALL-QUALITY-P, ALL-QUALITY-N, ALL-QUALITY-NEU,
  ALL-STYLE&OPTIONS-P, ALL-STYLE&OPTIONS-N, ALL-STYLE&OPTIONS-NEU,
  ALL-MISCELLANEOUS-P, ALL-MISCELLANEOUS-N, ALL-MISCELLANEOUS-NEU }
```

La idea de este método es producir todas las posibles opiniones construidas en base a una entidad, sus posibles aspectos y la polaridad que cada aspecto puede tomar. Luego, representarlas simplemente como un clase binaria que tiene como valor 1 si esta opinión está presente y 0 si no.

Entonces, en base al conjunto de etiquetas definido en el método podemos representar el corpus de la siguiente manera:

Textos	Categorías	GENERAL-P	GENERAL-N	GENERAL-NEU	...	PRICES-P	PRICES-N	PRICES-NEU
text ₁	SERVICE	1	0	0	...	0	0	0
text ₁	RESTAURANT	1	0	0	...	1	0	0
text ₂	LOCATION	1	0	0	...	0	0	0

FIGURA 3.9: 1.^{er} método para ABSA: Binarización en 3.8

Donde se eliminó el prefijo ALL- de cada etiqueta ya que se asume que la entidad ya está identificada por ser la única en el dominio.

Luego se puede aplicar o bien un algoritmo de clasificación binaria por clase de salida o un algoritmo multi-label tomando como etiquetas a todo el conjunto en 3.2.

Esta técnica, en la cual se descompone una clase objetivo, que toma valores en el conjunto potencia de una serie de etiquetas; en una serie de clases binarias que toman valor 1 si esa etiqueta está en el conjunto y 0 si no, se denomina Binary Relevance [9].

La gran desventaja de este método es el número de etiquetas de salida, en este caso, como se puede notar, existen 15 clases. Esto representa un problema al algoritmo de clasificación ya que tendrá con lidiar con el problema de overfitting sobre las clases que no aparecen tan a menudo.

Nosotros tuvimos este problema al utilizar este primer método, básicamente el algoritmo clasificaba tan bien las clases que no aparecen frecuentemente ya que les

asignaba 0 en todos los casos. El correcto balanceo del corpus supone una propiedad clave a la hora de clasificar mediante este método.

Método 2

El segundo método consiste en separar el problema de asignar un conjunto de etiquetas a un texto dado en n problemas de clasificación y utilizar la salida de un problema como entrada en el otro. Es decir, dado el corpus de entrada, realizar una clasificación para identificar primero las entidades. Luego utilizar ese resultado para poblar un corpus y deducir los aspectos y por último utilizar los últimos dos resultados para inferir la orientación del sentimiento.

Entonces, partiendo del corpus de opiniones inicial:

Textos	Categorías	Opiniones
text ₁	SERVICE	{(ALL, GENERAL, P)}
text ₁	RESTAURANT	{(ALL, GENERAL, P), (ALL, PRICES, P)}
text ₂	LOCATION	{(ALL, GENERAL, P)}

FIGURA 3.10: Corpus para ABSA: Restaurant Reviews

Construimos un nuevo corpus utilizando Binary Relevance (BR) para las entidades:

Textos	Categorías	ALL
text ₁	SERVICE	1
text ₁	RESTAURANT	1
text ₂	LOCATION	1

FIGURA 3.11: 2.^{do} método para ABSA: BR para entidades en 3.10

Obsérvese que como tenemos una única entidad, el sistema asume que ya está predeterminada y no es necesario realizar la clasificación.

El siguiente paso es realizar una segunda clasificación utilizando las entidades inferidas anteriormente como feature de entrada y aplicando Binary Relevance a los aspectos como salida. La siguiente tabla 3.12 se corresponde con el estado del corpus hasta el momento:

Textos	Categorías	Entidades	GENERAL	PRICES	QUALITY	...	MISCELLANEOUS
text ₁	SERVICE	ALL	1	0	0	...	0
text ₁	RESTAURANT	ALL	1	1	0	...	0
text ₂	LOCATION	ALL	1	0	0	...	0

FIGURA 3.12: 2.^{do} método para ABSA: BR para aspectos en 3.10

En esta etapa, los aspectos de los textos son identificados, y así como en el análisis de entidades, puede pasar que un texto tenga más de un aspecto para una entidad determinada. Esto se traduce en más filas de entrada en el corpus para la clasificación de la polaridad. Es decir, luego de este análisis, utilizamos los aspectos identificados y los agregamos como features de entrada como información para la clasificación de objetivos de opinión y construimos el siguiente corpus 3.13,

Donde como se puede notar, el corpus creció de tamaño al agregar la tercera fila por el aspecto PRICES de la segunda fila del corpus de la clasificación de aspectos

Textos	Categorías	Aspectos	P	N	NEU
text ₁	SERVICE	GENERAL	1	0	0
text ₁	RESTAURANT	GENERAL	1	0	0
text ₁	RESTAURANT	PRICES	1	0	0
text ₂	LOCATION	GENERAL	1	0	0

FIGURA 3.13: 2.^{do} método para ABSA: BR para la polaridad en 3.10

3.12. Aquí estamos realizando análisis de sentimiento basado en el aspecto inferido en la etapa anterior.

Este modelo de clasificación en el cual se construyen n corpus que utilizan como entrada las predicciones de los clasificadores de problemas anteriores se denominan modelo de clasificación en cadena [69]. Además, el clasificador compuesto por todos los anteriores que produce la salida completa (el conjunto de opiniones en este caso), se denomina “Ensemble Classifier” [70]. Este enfoque nos permite mitigar el problema de overfitting al tener un algoritmo por cada problema de clasificación en comparación con el método anterior.

Finalizando, nos queda responder cómo los clasificadores en cadena funcionan para producir la salida final. La idea es relativamente simple, si llamamos $\mathcal{L}_{\text{entity}}$, $\mathcal{L}_{\text{aspect}}$, $\mathcal{L}_{\text{polarity}}$, a los clasificadores para la entidad, aspecto y polaridad respectivamente; entonces partiendo de la figura tenemos:

$$\begin{aligned}\mathcal{L}_{\text{entity}}(\text{text}_1, \text{RESTAURANT}) &= \{\text{ALL}\} \\ \mathcal{L}_{\text{aspect}}(\text{text}_1, \text{RESTAURANT}, \text{ALL}) &= \{\text{GENERAL}, \text{PRICES}\} \\ \mathcal{L}_{\text{polarity}}(\text{text}_1, \text{RESTAURANT}, \text{ALL}, \text{GENERAL}) &= \text{P} \\ \mathcal{L}_{\text{polarity}}(\text{text}_1, \text{RESTAURANT}, \text{ALL}, \text{PRICES}) &= \text{P}\end{aligned}$$

Donde finalmente, la terna (entity, aspect, polarity) para un texto determinado se produce simplemente ejecutando los clasificadores en el orden de la terna. Es decir, si \mathcal{L} es el algoritmo que dado un texto y su categoría produce la terna (entity, aspect, polarity), utilizando este método el mismo se define mediante:

$$\mathcal{L}(\text{text}, \text{category}) = \bigcup_{e \in \mathcal{L}_{\text{entity}}(\text{text}, \text{category})} \bigcup_{a \in \mathcal{L}_{\text{aspect}}(\text{text}, \text{category}, e)} \mathcal{L}_{\text{polarity}}(\text{text}, \text{category}, e, a)$$

Donde no es más que dos recorridas, la primera sobre el conjunto de entidades inferidas $\mathcal{L}_{\text{entity}}$ y la segunda sobre el conjunto de aspectos inferidos $\mathcal{L}_{\text{aspect}}$ dado la entidad. Luego simplemente se aplica $\mathcal{L}_{\text{polarity}}$ basado en el par (entity, aspect) determinado. Justamente es por esto que el método se denomina clasificación en cadena, ya que se utiliza la salida del algoritmo inmediatamente anterior para entrenar el actual. Obsérvese que el único clasificador que no es multi-label es el de polaridad $\mathcal{L}_{\text{polarity}}$, ya que una vez que se determina el objetivo de opinión (entity, aspect); la polaridad asociada es única ya que un aspecto no puede tener más de una orientación del sentimiento.

En el siguiente capítulo vamos a describir los métodos desarrollados para construir \mathcal{L} basado en el modelo de clasificación en cadena.

Capítulo 4

Algoritmos de aprendizaje y recursos utilizados

En este capítulo vamos a exponer los detalles de los algoritmos de aprendizaje desarrollados para realizar la clasificación así como los recursos externos utilizados. Como lo habíamos adelantado en el capítulo anterior, los algoritmos implementados están diseñados para realizar la clasificación basado en el método de clasificadores en cadena 3.4, en el cual, se utiliza la salida de algoritmos ya entrenados como features en el corpus de entrenamiento del algoritmo a entrenar.

4.1. Algoritmos de clasificación

Esta sección define dos algoritmos, el primero es la línea base (baseline) utilizada para contrastar la performance del segundo algoritmo desarrollado para realizar la clasificación y tener buenas medidas de rendimiento. A continuación se expone la notación utilizada para representar el corpus partiendo de la salida del modulo de binarización para que el lector encuentre más sencillo el entendimiento del funcionamiento de los algoritmos.

La idea es elegir la representación del corpus de la siguiente manera:

Text	Features	Y_1	...	Y_k
t_1	f_{t_1}	1	...	0
t_2	f_{t_2}	0	...	1
\vdots	\vdots	\vdots	\vdots	\vdots
t_K	f_{t_K}	1	...	1

FIGURA 4.1: Conjunto de entrenamiento

Donde se ha simplificado el conjunto de sus características bajo **Features** y renombrado las clases de salida por $Y = \{Y_1, \dots, Y_k\}$ manteniendo el formato del conjunto de entrenamiento.

Obsérvese que cuando $Y = \text{ASPECTS}$, es decir $Y_1 = \text{GENERAL}$, $Y_2 = \text{PRICES}$, $Y_3 = \text{QUALITY}$... $Y_k = \text{MISCELLANEOUS}$ y **Features** = Categorías, el corpus representado es el utilizado para la clasificación de aspectos,

Textos	Categorías	GENERAL	PRICES	QUALITY	...	MISCELLANEOUS
text ₁	SERVICE	1	0	0	...	0
text ₁	RESTAURANT	1	1	0	...	0
text ₂	LOCATION	1	0	0	...	0

FIGURA 4.2: 2.^{do} método para ABSA: BR para aspectos en 3.10

Donde nuevamente, los textos del corpus son los siguientes:

text₁ = Servicio impecable y se come de maravilla a buen precio!

text₂ = Nos pusieron en la terraza, sin avisar previamente.

De la misma manera podemos representar el corpus para el cálculo de la orientación del sentimiento. La idea detrás de esta representación es construir un algoritmo de aprendizaje que se adapte al formato del corpus, porque luego podemos reutilizar su estructura tanto para el cálculo de entidades como de aspecto y polaridad.

Línea Base

La línea base consiste en la clasificación mediante las palabras más comunes que aparecen en los textos para una etiqueta de salida. Entonces, supongamos que la salida del preprocesamiento de los textos se traduce en una función que dado un texto, retorna la lista de tokens de acuerdo a las actividades realizadas en esa etapa.

Lo representamos mediante una función preprocessing que cuando toma un texto del corpus, por ejemplo text₁, retorna lo siguiente:

preprocessing(text₁) = [servicio, impecable, come, maravilla, buen, precio]

Luego dado un (text, feature) como entrada en el corpus, construimos el conjunto de las $n \in \mathbb{N}$ palabras más comunes sobre los datos de entrenamiento que están clasificadas con la etiqueta $y \in \{Y_1, \dots, Y_k\}$ y que contienen la feature feature.

Es así que, por ejemplo, para el corpus 4.2, utilizamos el conjunto de relaciones del corpus de restaurantes en 2.4.1 para calcular las dos palabras más comunes para cada par categoría y aspecto. La siguiente tabla simplemente ilustra lo almacenado por el algoritmo para el aspecto PRICES:

Categories Aspects	Most-Common-PRICES
RESTAURANT	[precio, caro]
FOOD	[menú, plato]
DRINKS	[trago, bebida]
AMBIENCE	∅
SERVICE	∅
LOCATION	∅

Luego de construir esta tabla para cada aspecto en el sistema el algoritmo asigna una etiqueta al texto siempre y cuando contenga una palabra más común luego de ser procesado. En este ejemplo, el algoritmo infiere etiquetar a text₁ con PRICES ya que, simplemente, compara la lista de las dos palabras más comunes en el corpus que contienen al aspecto y como ve que «precio» es una de las palabras más comunes que está en el texto, decide etiquetarla como 1. Análogamente, el mismo

cálculo se realiza con el resto de los aspectos. Vamos a llamar a este estimador E_y donde y es una etiqueta de salida (por ejemplo E_{PRICES} para PRICES). La línea base es el algoritmo que consiste en calcular este estimador para cada etiqueta de salida, es decir:

$$\mathcal{L} = (E_{Y_1}, \dots, E_{Y_k})$$

Nuevamente, si queremos calcular \mathcal{L} sobre 4.2 para inferir los aspectos de text_1 para la categoría RESTAURANT, deberíamos hacer el siguiente cálculo:

$$\begin{aligned} \mathcal{L}(\text{text}_1, \text{RESTAURANT}) = & (E_{GENERAL}(\text{text}_1, \text{RESTAURANT}), \\ & E_{PRICES}(\text{text}_1, \text{RESTAURANT}), \\ & \dots, \\ & E_{MISCELLANEOUS}(\text{text}_1, \text{RESTAURANT})) = (1, 1, \dots, 0) \end{aligned}$$

Algoritmo propuesto

Vamos a explicar el funcionamiento del algoritmo propuesto partiendo del corpus de entrenamiento de aspectos 4.2. Como observación vamos a mencionar que la estructura del algoritmo funciona también mediante el método de clasificadores en cadena. Es decir, vamos a definir una estructura general para luego acomodar las features y los atributos de salida para la clasificación de entidades, aspectos y polaridades.

El diagrama 4.1 ilustra la arquitectura desarrollada para realizar la clasificación en el corpus 4.1 mediante redes neuronales artificiales:

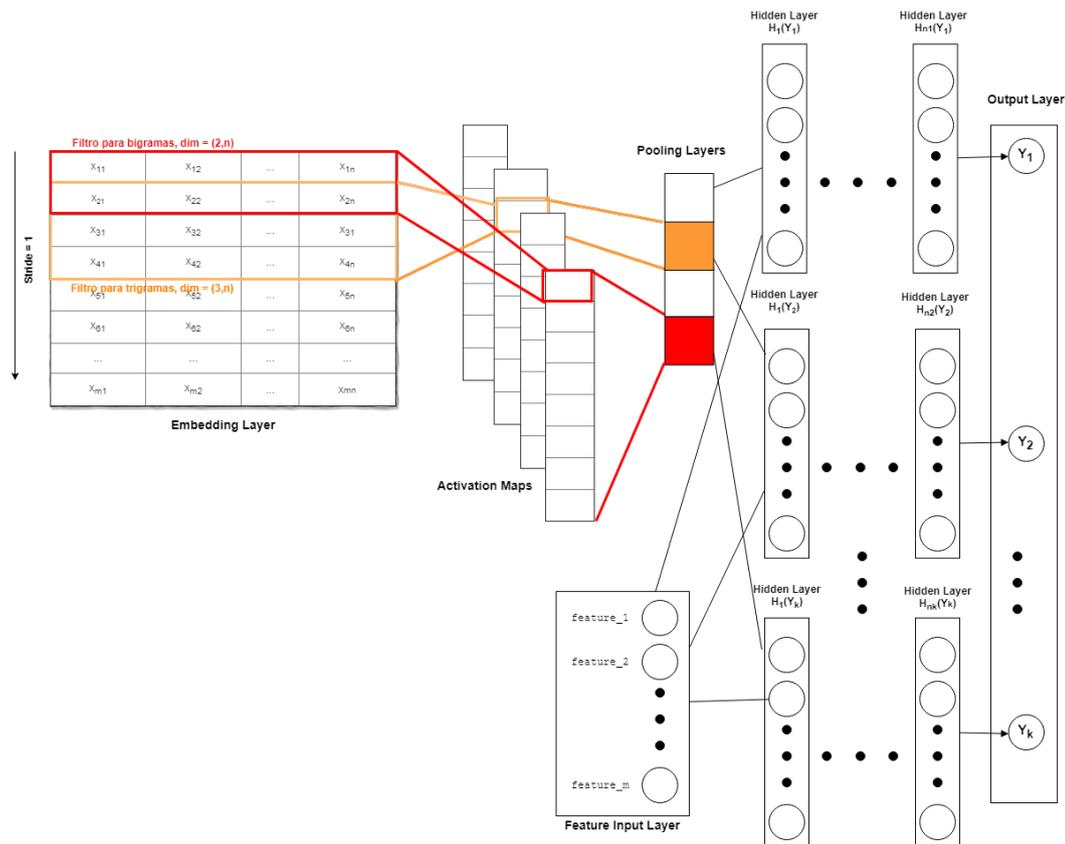


FIGURA 4.3: Multi-Output CNN para ABSA

La CNN fue inspirada en la red desarrollada por Yoon Kim [94] para la clasificación de textos, y que ha demostrado ser bastante competitiva, en términos de desempeño sobre distintos corpus en la literatura.

El algoritmo comienza con la inicialización de la capa de entrada denominada **Embedding Layer**, la cual consiste en la representación de los textos de los documentos en una lista de vectores de palabras. La idea es la siguiente: tomamos los textos de los documentos en el corpus 4.2 y hacemos uso nuevamente de la función preprocessing para obtener la lista de palabras luego del procesamiento de los textos:

```
preprocessing(text1) = [servicio, impecable, come, maravilla, buen, precio]
preprocessing(text2) = [pusieron, terraza, avisar, previamente]
```

A esto se le aplica una nueva función denominada vectorizer la cual consiste en la aplicación punto a punto de la función vocabulary 3.1 sobre la lista de palabras anteriores:

```
vectorizer([servicio, impecable, come, maravilla, buen, precio]) = [1, 2, 3, 4, 5, 6]
vectorizer([pusieron, terraza, avisar, previamente]) = [7, 8, 9, 10, 0, 0]
```

Obsérvese que como este es un ejemplo ilustrativo, no existen palabras repetidas, por lo tanto, todas tienen un número de identificador ascendente. Además, notar que se relleno con ceros a la derecha (**Zero Padding**) la vectorización de text₂ luego del procesamiento, para que la dimensión del vector de salida de la función vectorizer

coincida con el largo m de la lista más larga. Esto se utiliza para obtener la dimensión de la **Embedding Layer**, la cual como se puede ver en 4.1, consiste en m filas por n columnas, siendo n el largo de los vectores de palabras.

La inicialización de esta capa termina cargando los valores de los vectores de palabras para todas las palabras del vocabulario. Esto se hace almacenando los vectores de palabras precargados 3.3 en la capa de vectores de palabras, para que la misma pueda decidir cuál es el valor de una palabra dado su índice en el vocabulario.

Ejemplificando esta etapa, en tiempo de entrenamiento la acción de la capa de vectores de palabras es la siguiente, se le presenta un texto, por ejemplo text_1 , se efectúan dos operaciones `preprocessing` y `vectorizer`; y se tiene una lista de índices en el vocabulario seguida por rellenos de ceros (si aplica)

$$\text{EmbeddingLayer}([1, 2, 3, 4, 5, 6]) = [\text{embedding}(1), \text{embedding}(2), \dots, \text{embedding}(6)]$$

$$\text{EmbeddingLayer}([1, 2, 3, 4, 5, 6]) = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{61} & x_{62} & \dots & x_{6n} \end{pmatrix}$$

Obsérvese que esta matriz es exactamente la misma que la presentada en el diagrama para el texto de entrada text_1 . Cuando este es alimentado como entrada en la red, pasa por este proceso antes de empezar a hacer cálculos.

Utilizamos vectores de palabras pre-entrenados, es decir, construidos previamente y cargados en el sistema a posteriori. La tabla 4.4 lista todos los vectores de palabras utilizados así como su algoritmo y las referencias apropiadas:

Corpus	Algoritmo	Referencia
Spanish Billion Words Corpus ¹	Word2Vector (Skip N-gram)	[14, 55]
Spanish Unannotated Corpora (SUC) ²	FastText	[1]
Spanish Billion Words Corpus	Glove	[30]
Spanish Wikipedia Dump ³	Glove	[50]

FIGURA 4.4: Recursos utilizados

Luego, como lo vimos en 2.5.3 podemos aplicar la fórmula de convolución bidimensional para extraer características de los mismos. Esta red queda definida mediante los siguientes hiper-parámetros 2.5.3:

¹<https://crscardellino.github.io/SBWCE/>

²<https://github.com/josecannete/spanish-corpora>

³<http://dumps.wikimedia.org/eswiki/20150228/eswiki-20150228-pages-articles.xml.bz2>

Stride	Zero-Padding	Filters		
		Depth	Size	Activation
1	No	600	(2, embedding_dim)	ReLU
		600	(3, embedding_dim)	ReLU
		600	(4, embedding_dim)	ReLU
		600	(5, embedding_dim)	ReLU

FIGURA 4.5: Hiperparámetros en la red.

Donde `embedding_dim` es el tamaño de los vectores de palabras. Obsérvese que como ya rellenamos con ceros a la derecha la secuencia de entrada a la red, y la dimensión de salida es la que queremos, no es necesario realizarlo en la convolución, justamente por eso **Zero-Padding** = 0. El parámetro `Depth` = 600 se determinó en base a pruebas realizadas sobre corpus de análisis de sentimiento y se decidió utilizarlo para este problema. Lo ideal es programar este parámetro en el sistema para que pueda ser ajustado según ciertas heurísticas.

Entonces la idea es la siguiente, como indica 4.5, primero en la convolución vamos a aplicar una serie de 600 filtros de tamaño $(2, \text{embedding_dim})$ con función de activación *ReLU* avanzando el filtro de a un paso (*Stride* = 1). La figura 4.1 ilustra este proceso, como se puede notar vamos a ir tomando pares de vectores de palabras dentro de la **Embedding Layer**, realizando la construcción e ir moviendo el filtro un paso en el orden de la secuencia de palabras preprocesadas en el texto. Esto se traduce en ir capturando características para bigramas de dentro del texto.

Luego de la convolución obtenemos un conjunto de mapas de características, **Activation Maps**, al cual se les aplica una función denominada **Max Pooling** que consiste en tomar el valor máximo de cada vector de activación y construir una capa en la CNN con este valor, denominada **Pooling Layer** como se puede ver en la figura 4.1.

Este vector se concatena con la capa de características (**Feature Layer**), la cual contiene el one-hot encoding de las features del sistema, y toda esta información se la propaga a k redes neuronales densas disjuntas que atacaran el problema de clasificación binaria sobre las etiquetas $\{Y_1, \dots, Y_k\}$. Las mismas son una serie de perceptrones multicapa con función de activación tangente hiperbólica. Finalmente la capa de salida se compone por un conjunto de perceptrones con función de activación sigmoideal.

La red neuronal artificial utiliza la función de entropía binaria cruzada como función de costo entre la predicción \hat{y} y el valor real y

$$-\frac{1}{n} \sum_{i=1}^n y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$

La idea detrás de esta decisión es que como las salidas del algoritmo son funciones sigmoideas, no se rigen por una distribución de probabilidad por tanto, la asignación de etiquetas por cada neurona de salida no afecta las decisiones de otras neuronas, lo cual se traduce en que el algoritmo elige el conjunto de etiquetas para una entrada asumiendo que cada etiqueta es distinta e independiente entre sí. Esto podría generar inconsistencias al realizar el cálculo de la polaridad, ya que podemos asumir que existe cierta distribución de probabilidad entre las etiquetas; sin embargo, como lo sugieren las pruebas en el capítulo 5, el algoritmo logra capturar esta relación de dependencia.

Obsérvese que, como dijimos, el tamaño de los filtros aplicados en la Embedding Layer se ajustan en largo, a los vectores de palabras utilizados. Cuando aplicamos el filtro (q , `embedding_dim`) donde $q = 1 \dots 4$ lo que estamos haciendo es calcular los q -gramas para cada palabra representada por el vector de palabra esperando a que un filtro de los 600 aplicados capture esta información para que sea luego propagada hacia adelante en la red.

En la práctica, al ser los corpus chicos en tamaño, utilizamos solamente una capa oculta, sin embargo la arquitectura de la red neuronal es flexible en cuanto a la cantidad de capas ocultas aplicables por etiquetas de salida.

El siguiente capítulo es el encargado de realizar la evaluación del sistema comparando estos dos enfoques.

Capítulo 5

Evaluación del sistema propuesto

En este capítulo presentamos el desempeño del algoritmo de acuerdo a una serie de métricas. Las mismas son definidas brevemente en la siguiente sección y luego la próxima sección es la encargada de ilustrar los resultados obtenidos.

5.1. Métricas utilizadas

Las siguientes son las métricas utilizadas para evaluar el rendimiento de los dos algoritmos propuestos. Dado que el conjunto de salidas está modelado por una serie de $k \in \mathbb{N}$ variables $\{Y_1, \dots, Y_k\}$ binarias, las métricas propuestas evalúan a cada etiqueta por separado. Entonces dado Y_i , definimos las siguientes variables:

tp_i = Cantidad de ejemplos de entrenamiento de clase $Y_i = 1$ clasificados correctamente.

tn_i = Cantidad de ejemplos de entrenamiento de clase $Y_i = 0$ clasificados correctamente.

fn_i = Cantidad de ejemplos de entrenamiento de clase $Y_i = 1$ clasificados erróneamente.

fp_i = Cantidad de ejemplos de entrenamiento de clase $Y_i = 0$ clasificados erróneamente.

Si \hat{Y}_i es la predicción del algoritmo para la clase Y_i , definimos las siguientes métricas de desempeño para clasificadores binarios:

$$\begin{aligned} \text{Accuracy (Acc)} &= \mathbb{P}(Y_i = \hat{Y}_i) &= \frac{tp_i + tn_i}{tp_i + tn_i + fp_i + fn_i} \\ \text{Recall (R)} &= \mathbb{P}(\hat{Y}_i = 1 | Y_i = 1) &= \frac{tp_i}{tp_i + fn_i} \\ \text{Precision (P)} &= \mathbb{P}(Y_i = 1 | \hat{Y}_i = 1) &= \frac{tp_i}{tp_i + fp_i} \\ \text{F}_1 \text{ Score} &= &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned}$$

5.2. Análisis de los resultados

A continuación presentamos los resultados obtenidos en la identificación de aspectos del sistema. El sistema realizó la clasificación utilizando todos los vectores de palabras pre-entrenados en 4.4 pero debido al tamaño de los corpus (alrededor de dos mil textos) no se pudo notar la influencia de los mismos sobre las métricas, de forma tal que para evaluar la red artificial 4.1 solamente se utilizó un vector de palabras en particular.

Resultados sobre corpus de SemEval

En las próximas secciones mostramos los análisis realizados sobre los corpus de SemEval. Primero se realiza un análisis de desempeño en la densificación de los aspectos en el corpus y luego analizamos los resultados en la tarea de análisis de sentimientos basado en los aspecto recuperados.

Identificación de aspectos para Restaurant Reviews en español

Las siguiente tabla contiene los resultados obtenidos por el clasificador de palabras más comunes (base) y la red artificial para la tarea de identificación de aspectos del sistema utilizando el método número dos 3.4 sobre el corpus “Restaurant Reviews” 2.4.1 en español.

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
GENERAL	58.18 %	73.41 %	64.91 %	83.58 %	88.89 %	86.15 %	252.0
MISCELLANEOUS	0.73 %	33.33 %	1.43 %	0.00 %	0.00 %	0.00 %	3.0
PRICES	12.33 %	88.10 %	21.64 %	62.00 %	73.81 %	67.39 %	42.0
QUALITY	36.33 %	79.56 %	49.89 %	81.51 %	70.80 %	75.78 %	137.0
STYLE_OPTIONS	10.55 %	78.38 %	18.59 %	43.75 %	37.84 %	40.58 %	37.0

FIGURA 5.1: Identificación de aspectos para Restaurant Reviews 2.4.1

A primera vista notamos el desbalance del corpus en términos de cantidad de ejemplos contenidos en cada clase. Sobre un total de 471 ejemplos de evaluación tomados de forma aleatoria, las clases MISCELLANEOUS y STYLE_OPTIONS tienen una muestra de solamente 3 y 37 respectivamente (ver “support”). Y la masa de ejemplos se la lleva GENERAL con 252.

Esto viola la primera norma de las tres propiedades deseables en un corpus vista en 3.3, ya que el corpus al no ser representativo de todo el espacio impacta en la performance de los algoritmos.

En particular, para aquellas etiquetas que tengan menor masa sobre el total, la cantidad de ejemplos positivos que se puedan convertir en falsos negativos es chica mientras que la cantidad de ejemplos negativos que se puedan convertir en falsos positivos es alta. De manera que, debido a la naturaleza del corpus, la línea base tiene alta recuperación (recall) pero baja precisión (precision). Además, para el clasificador de palabras más comunes, la medida F no supera el 65 % entre todas las etiquetas.

Destacamos la capacidad de la red neuronal para obtener buena performance cuando el corpus no cumple los requisitos que tiene que tener un buen conjunto de datos. Es decir, si miramos la etiqueta GENERAL podemos ver el corpus medianamente balanceado con 252 ejemplos positivos en support produce una medida F del

86,15 %. En otras palabras, si bien el corpus es chico y desbalanceado, la red neuronal realiza bien la clasificación para aquellas etiquetas que se llevan gran parte de la masa de ejemplos.

Obsérvese que para la clase PRICES, con tan solo el 8,9 % de los ejemplos positivos, obtiene una medida F del 67 %, es decir, aprende a clasificar ejemplos de entrenamiento sobreponiéndose a las condiciones del corpus. Más aún, realizamos una prueba utilizando el algoritmo sobre el texto «comida cara pero buena» y retorno como salida {PRICES, QUALITY}, lo cual habla de que el sistema puede inferir asociaciones entre palabras de muchas categorías sobre el texto.

Identificación de aspectos para Restaurant Reviews en inglés

Los resultados obtenidos por el clasificador de palabras más comunes (base) y la red artificial para la tarea de identificación de aspectos del sistema utilizando el método número dos 3.4 sobre el corpus “Restaurant Reviews” 2.4.1 en inglés son los siguientes:

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
GENERAL	56.18 %	72.27 %	63.22 %	78.41 %	80.91 %	79.64 %	220.0
MISCELLANEOUS	5.86 %	62.50 %	10.71 %	28.57 %	8.33 %	12.90 %	24.0
PRICES	15.32 %	92.68 %	26.30 %	60.61 %	48.78 %	54.05 %	41.0
QUALITY	38.08 %	81.68 %	51.94 %	67.61 %	73.28 %	70.33 %	131.0
STYLE_OPTIONS	10.78 %	80.65 %	19.01 %	35.00 %	22.58 %	27.45 %	31.0

FIGURA 5.2: Identificación de aspectos para Restaurant Reviews 2.4.1 en inglés

Observamos aquí que como la distribución de masa entre las etiquetas del corpus es similar a su versión en español, la clasificación de la red es similar. Lo cual habla de la adaptabilidad de la red al idioma simplemente adaptando los vectores de palabras al idioma del corpus. Además podemos notar que la red tiene mejor desempeño que la línea base ya que la medida F es superada por la red en todas las etiquetas.

Identificación de aspectos para Laptops

A continuación, los resultados sobre el corpus “Laptops” 2.4.1 son los siguientes:

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
CONNECTIVITY	7.94 %	90.91 %	14.60 %	66.67 %	18.18 %	28.57 %	11.0
DESIGN_FEATURES	17.24 %	73.53 %	27.93 %	56.14 %	47.06 %	51.20 %	68.0
GENERAL	49.29 %	74.19 %	59.23 %	71.01 %	64.52 %	67.61 %	186.0
MISCELLANEOUS	12.24 %	78.38 %	21.17 %	46.15 %	16.22 %	24.00 %	37.0
OPERATION_PERFORMANCE	27.65 %	86.17 %	41.86 %	67.57 %	53.19 %	59.52 %	94.0
PORTABILITY	4.61 %	100.00 %	8.81 %	80.00 %	40.00 %	53.33 %	10.0
PRICE	9.84 %	100.00 %	17.91 %	76.47 %	54.17 %	63.41 %	24.0
QUALITY	27.24 %	68.87 %	39.04 %	71.08 %	55.66 %	62.43 %	106.0
USABILITY	10.55 %	80.56 %	18.65 %	45.24 %	52.78 %	48.72 %	36.0

FIGURA 5.3: Identificación de aspectos para Laptops 2.4.1

Donde las columnas sobre la izquierda se corresponden las métricas obtenidas por el clasificador de palabras más comunes (base) y la de las derecha por la red artificial.

Obsérvese que de los 572 ejemplos de evaluación, las etiquetas GENERAL, QUALITY y OPERATION_PERFORMANCE se llevan el 67.5 % de la masa correspondientes a 386

ejemplos, lo cual habla del desbalance del corpus.

Nuevamente se cumple que por la naturaleza del corpus, la recuperación supere como medida a la precisión para el clasificador de palabras más comunes, sin embargo la red neuronal supo sobrellevar este problema. La etiqueta PRICES con tan solo 24 ejemplos logra obtener una medida F de 63,4 % y la etiqueta PORTABILITY logra pasar de una precisión base de 4,61 % a 80 % por parte de la red artificial con tan solo 10 ejemplos. Esto nos hace concluir que la red puede generalizar muy bien si se le suministran ejemplos de entrenamiento generales para todo el espacio.

Análisis de sentimientos para Restaurant Reviews en español

Para evaluar el sistema en la tarea de análisis de sentimientos basados en aspectos sobre corpus de SemEval, primero utilizamos la lista de aspectos en el gold standard del corpus; luego realizamos la clasificación en dos pasos: primero utilizamos el sistema para descubrir los aspectos y por último los utilizamos como datos de entrada en el corpus para realizar la clasificación del sentimiento.

Las siguiente tabla contiene los resultados obtenidos por el clasificador para la tarea de análisis de sentimientos basado en aspectos utilizando el método [numero dos 3.4](#) sobre el corpus “Restaurant Reviews” [2.4.1](#) en español.

	precision (gold)	recall (gold)	f1-score (gold)	precision	recall	f1-score	support
P	87.44 %	89.27 %	88.34 %	77.51 %	76.70 %	77.11 %	382.0
NEU	20.00 %	8.70 %	12.12 %	0.00 %	0.00 %	0.00 %	23.0
N	67.27 %	56.06 %	61.16 %	38.20 %	25.76 %	30.77 %	132.0

FIGURA 5.4: Análisis de sentimientos para Restaurant Reviews [2.4.1](#)

Obsérvese que pasamos de un conjunto de testeo de 471 ejemplos (ver *support* en [5.1](#)) a un conjunto de 537 ejemplos. Esto es debido al algoritmo utilizado [3.4](#) que agrega al corpus el par (text, aspect) para cada aspecto en el texto.

Nuevamente notamos que la falta de balanceo afecta significativamente a la performance en la clasificación. Notar que la clase NEU, es decir, la polaridad neutral, solamente tiene 23 ejemplos sobre 537 posibles. Debido a esto la red aprende a no etiquetar NEU en la mayoría de los casos luego las métricas de Precisión y Recall son bajas o se anulan. Este es un ejemplo que muestra cómo la “Accuracy” por sí sola, no es una buena medida de rendimiento. El sistema tiene una Accuracy de más de 90 % por que clasifica todos los ejemplos negativos en forma correcta ya que la distribución de la clase no es buena. Luego tn_i y fp_i son los únicos elementos que influyen en la “Accuracy” y la métrica termina siendo cercana a la unidad. Pero sin embargo esta métrica esconde el hecho de que todos los ejemplos son 0 y el algoritmo solo aprendió a memorizar este resultado.

Se observa además que las métricas decrementaron de acuerdo al problema que se está planteando. Este nuevo sistema arrastra los errores del clasificador de aspectos para luego agregar la salida al corpus de análisis de sentimiento, por lo cual, si existe un error en el primero, el mismo va a ser propagado sobre el segundo.

Análisis de sentimientos para Restaurant Reviews en inglés

Nuevamente, anotamos con “gold” a la evaluación utilizando la lista de aspectos en el gold standard del corpus y luego realizamos la clasificación con la predicción de los aspectos. La siguiente tabla contiene los resultados obtenidos por el clasificador sobre el corpus “Restaurant Reviews” en inglés:

	precision (gold)	recall (gold)	f1-score (gold)	precision	recall	f1-score	support
P	76.44 %	90.40 %	82.84 %	64.38 %	78.33 %	70.67 %	323.0
NEU	22.22 %	9.52 %	13.33 %	0.00 %	0.00 %	0.00 %	21.0
N	70.18 %	52.63 %	60.15 %	26.61 %	19.08 %	22.22 %	152.0

FIGURA 5.5: Análisis de sentimientos para Restaurant Reviews [2.4.1](#) en inglés

En la misma podemos visualizar exactamente las conclusiones del análisis pasado. Obsérvese que la precisión en la clase N tuvo una baja de más de la mitad y la recall decremento 33.5 puntos. Esto es debido a que tenemos 323 ejemplos para la clase P y 152 ejemplos para la clase N, luego el algoritmo aprende a generalizar sobre la clase de más masa en el conjunto de entrenamiento y gracias a esto se observa que el rendimiento en P es muy superior a los demás.

Análisis de sentimientos para Laptops

A continuación, los resultados sobre el corpus “Laptops” son los siguientes:

	precision (gold)	recall (gold)	f1-score (gold)	precision	recall	f1-score	support
P	81.71 %	85.89 %	83.75 %	64.10 %	72.37 %	67.98 %	333.0
NEU	32.14 %	27.27 %	29.51 %	0.00 %	0.00 %	0.00 %	33.0
N	80.77 %	60.87 %	69.42 %	56.07 %	28.99 %	38.22 %	207.0

FIGURA 5.6: Análisis de sentimientos para Laptops [2.4.1](#)

Nuevamente, el hecho más notorio aquí es la capacidad de la red de obtener buenas medidas de rendimiento para las clases bien balanceadas. Observar que la clasificación de NEU no se desempeñó muy bien ya que la cantidad de ejemplos fue pequeña en comparación con las otras clases. La reducción de las medidas de rendimiento en comparación con el gold es esperable ya que la red arrastra errores de clasificaciones anteriores.

Resultados sobre corpus de TASS

A continuación los resultados obtenidos para los corpus de TASS.

Identificación de entidades para Social TV

Las siguiente tabla contiene los resultados obtenidos por el clasificador de palabras más comunes (base) y la red artificial para la tarea de identificación de entidades del sistema utilizando el método número dos 3.4 sobre el corpus “Social TV” 2.4.2.

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
Afición	10.18 %	90.32 %	18.30 %	76.47 %	41.94 %	54.17 %	62.0
Alexis_Sánchez	0.54 %	100.00 %	1.08 %	50.00 %	100.00 %	66.67 %	1.0
Alvaro_Arbeloa	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	1.0
Andrés_Iniesta	0.72 %	100.00 %	1.44 %	0.00 %	0.00 %	0.00 %	1.0
Angel_Di_María	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Arbitro	0.98 %	75.00 %	1.94 %	0.00 %	0.00 %	0.00 %	4.0
Asier_Illarramendi	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Atlético_de_Madrid	0.34 %	100.00 %	0.67 %	0.00 %	0.00 %	0.00 %	1.0
Autoridades	1.95 %	88.89 %	3.82 %	0.00 %	0.00 %	0.00 %	9.0
Barcelona	16.16 %	98.84 %	27.78 %	47.57 %	56.98 %	51.85 %	86.0
Carles_Puyol	1.48 %	100.00 %	2.91 %	0.00 %	0.00 %	0.00 %	3.0
Cesc_Fàbregas	0.89 %	100.00 %	1.76 %	0.00 %	0.00 %	0.00 %	2.0
Cristiano_Ronaldo	4.26 %	96.00 %	8.16 %	52.00 %	52.00 %	52.00 %	25.0
Dani_Alves	1.36 %	100.00 %	2.68 %	0.00 %	0.00 %	0.00 %	4.0
Dani_Carvajal	1.00 %	100.00 %	1.98 %	33.33 %	50.00 %	40.00 %	2.0
Entrenador	5.00 %	100.00 %	9.52 %	38.10 %	34.78 %	36.36 %	23.0
Equipo	0.25 %	50.00 %	0.50 %	0.00 %	0.00 %	0.00 %	2.0
Fábio_Coentrão	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Gareth_Bale	10.65 %	97.78 %	19.21 %	53.66 %	48.89 %	51.16 %	45.0
Iker_Casillas	3.85 %	100.00 %	7.42 %	64.71 %	64.71 %	64.71 %	17.0
Isco	3.49 %	100.00 %	6.74 %	50.00 %	66.67 %	57.14 %	3.0
Javier_Mascherano	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	1.0
James_Rodríguez	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	1.0
José_Manuel_Pinto	3.09 %	100.00 %	6.00 %	50.00 %	25.00 %	33.33 %	12.0
Jugador	2.04 %	91.67 %	4.00 %	0.00 %	0.00 %	0.00 %	12.0
Karim_Benzema	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Lionel_Messi	8.76 %	100.00 %	16.12 %	52.11 %	84.09 %	64.35 %	44.0
Luka_Modric	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Marc_Bartra	1.86 %	100.00 %	3.66 %	33.33 %	57.14 %	42.11 %	7.0
Neymar_Jr.	1.57 %	100.00 %	3.08 %	41.67 %	83.33 %	55.56 %	6.0
Otro_Equipo	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Otro_Jugador	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Partido	17.61 %	91.74 %	29.54 %	100.00 %	40.37 %	57.52 %	109.0
Pedro_Rodríguez	0.36 %	100.00 %	0.71 %	0.00 %	0.00 %	0.00 %	1.0
Pepe	1.29 %	100.00 %	2.54 %	0.00 %	0.00 %	0.00 %	3.0
Real_Madrid	24.29 %	98.55 %	38.97 %	66.67 %	42.03 %	51.56 %	138.0
Retransmisión	0.51 %	28.57 %	1.00 %	0.00 %	0.00 %	0.00 %	7.0
Sergio_Busquets	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Sergio_Ramos	11.40 %	100.00 %	20.47 %	79.59 %	81.25 %	80.41 %	48.0
Xabi_Alonso	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.0
Xavi_Hernández	0.30 %	100.00 %	0.60 %	0.00 %	0.00 %	0.00 %	1.0

FIGURA 5.7: Identificación de entidades para Social TV 2.4.2

Observamos que Partido, Afición, Real_Madrid y Barcelona son las clases que se llevan la mayor masa de ejemplos, todas por encima de las 60 unidades; debido a esto, la medida F de estas clases supera el 50%. Nuevamente tenemos el problema de desbalance en el corpus, razón por la cual el algoritmo de clasificación por palabras más comunes se desempeña muy bien en la medida de recuperación pero su precisión no supera el 25% en todas las clases. Sin embargo, notamos que la precisión mejora considerablemente en la red artificial, obsérvese que Iker_Casillas pasó de tener una precisión del 3.85% a una del 64.71% con tan solo 17 ejemplos sobre el total de 681, lo que se corresponde con un incremento del 60.86% sobre el 2.49% de la masa de evaluación. Esta mejora considerable se observa en prácticamente todas las etiquetas que superan las 20 unidades.

Identificación de entidades para STOMPOL

Los siguientes son los resultados del sistema sobre el corpus “STOMPOL” 2.4.2 para la identificación de entidades:

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
Ciudadanos	26.19 %	100.00 %	41.51 %	100.00 %	54.55 %	70.59 %	22.0
Izquierda_Unida	37.35 %	100.00 %	54.39 %	52.94 %	29.03 %	37.50 %	31.0
Partido_Popular	40.94 %	96.83 %	57.55 %	76.47 %	41.27 %	53.61 %	63.0
Partido_Socialista_Obrero_Espanol	35.96 %	100.00 %	52.90 %	68.57 %	58.54 %	63.16 %	41.0
Podemos	20.27 %	93.75 %	33.33 %	63.64 %	43.75 %	51.85 %	16.0
Union_Progreso_y_Democracia	33.33 %	91.30 %	48.84 %	89.47 %	73.91 %	80.95 %	23.0

FIGURA 5.8: Identificación de entidades para STOMPOL 2.4.2

Nuevamente observamos el gran desbalance del corpus donde el Partido_Popular y el Partido_Socialista_Obrero_Espanol se llevan la mayoría de ejemplos de entrenamiento, lo que nuevamente produce que el clasificador por palabras más comunes tenga buen rendimiento en la recuperación. Sin embargo, existe una anomalía, ya que por primera vez vemos a la red artificial tener menos performance que la línea base. Obsérvese que para la etiqueta de mayor masa Partido_Popular la medida F en la línea base es de 57.55 % mientras que en la red es de 53.61 %. La razón por la que ocurre esto es por que los textos del corpus fueron tomados de Twitter de forma tal que las cuentas “@populares”, “@PPopular” y “@PP” estaban siempre presente en los textos. Luego, como el clasificador de palabras más comunes no quitaba este carácter y así, estas cuentas aparecían en el top de la lista de palabras más comunes para esta etiqueta, lo que hacía clasificar todos los textos que lo contienen de forma correcta. La red tuvo que aprender esta relación y creemos que no estuvo tan fuera de rango respecto a la línea base. Este comportamiento se da en todas las etiquetas, debido a la naturaleza del corpus.

Identificación de aspectos para STOMPOL

STOMPOL es el único corpus dentro de la lista que además de identificación de entidades también requiere identificación de aspectos. La siguiente son las métricas obtenidas por la línea base y la red artificial:

	precision (base)	recall (base)	f1-score (base)	precision	recall	f1-score	support
Economia	9.78 %	94.74 %	17.73 %	55.56 %	26.32 %	35.71 %	19.0
Educacion	4.05 %	60.00 %	7.59 %	33.33 %	20.00 %	25.00 %	5.0
Otros_aspectos	9.57 %	94.74 %	17.39 %	16.67 %	10.53 %	12.90 %	19.0
Propio_partido	80.00 %	98.06 %	88.12 %	85.03 %	91.61 %	88.20 %	155.0
Sanidad	1.12 %	50.00 %	2.20 %	0.00 %	0.00 %	0.00 %	2.0

FIGURA 5.9: Identificación de aspectos para STOMPOL 2.4.2

De los 200 ejemplos de evaluación 78 % los contiene la etiqueta Propio_Partido. Nuevamente, como la mejor estrategia es basarse simplemente en las cuentas de Twitter y clasificar de forma positiva si la contiene, no es de esperar que la línea base le haya ido tan bien. Lo que sí podemos destacar, es que la red artificial haya conseguido capturar esta relación. Observar que para aspectos que no se deducen de cuentas de Twitter como lo puede ser Economía y Educación la red neuronal tuvo mejor desempeño que la línea base. Observar que los textos que están etiquetados con Otros_aspectos también contienen cuentas de Twitter asociadas como “@FeminismoDgital”, “@anguitista” o “@anguitista_siempre” entre otras, razón por la cual el clasificador de palabras más comunes se desempeñó mejor.

Análisis de sentimientos para Social TV

Para evaluar el sistema en la tarea de análisis de sentimientos basados en aspectos sobre corpus de TASS, primero utilizamos la lista de entidades (y aspectos para STOMPOL) en el gold standard del corpus; luego realizamos la clasificación en dos pasos: primero utilizamos el sistema para descubrir las entidades (y aspectos para STOMPOL) y por último los utilizamos como datos de entrada en el corpus para realizar la clasificación del sentimiento.

Las siguiente tabla contiene los resultados obtenidos por el clasificador para la tarea de análisis de sentimientos basado en aspectos utilizando el método numero dos 3.4 sobre el corpus "Social TV" 2.4.2.

	precision (gold)	recall (gold)	f1-score (gold)	precision	recall	f1-score	support
P	70.93 %	59.68 %	64.82 %	60.21 %	30.91 %	40.85 %	372.0
NEU	46.36 %	46.54 %	46.45 %	40.82 %	30.77 %	35.09 %	260.0
N	55.61 %	46.67 %	50.75 %	45.51 %	29.80 %	36.02 %	255.0

FIGURA 5.10: Análisis de sentimientos para Social TV 2.4.2

Obsérvese que las clases están medianamente balanceadas, pero debido a tener pocos ejemplos de entrenamientos y desbalance en las entidades, no se obtienen buenas métricas. Además la mayoría de textos en el corpus contienen eufemismos muy difíciles de capturar por la red.

Análisis de sentimientos para STOMPOL

Los resultados obtenidos por la red sobre el corpus "STOMPOL" 2.4.2 son los siguientes:

	precision (gold)	recall (gold)	f1-score (gold)	precision	recall	f1-score	support
P	76.47 %	22.81 %	35.14 %	16.67 %	1.75 %	3.17 %	57.0
NEU	28.17 %	28.99 %	28.57 %	34.38 %	47.83 %	40.00 %	69.0
N	65.24 %	68.93 %	67.03 %	58.96 %	57.63 %	58.29 %	177.0

FIGURA 5.11: Análisis de sentimientos para STOMPOL 2.4.2

Debido a que para este corpus utilizamos la clasificación anterior de entidades y aspectos, la el desempeño en el cálculo de la polaridad de vio muy afectado. Para la clase P la medida F se decremento 32 % puntos. Sin embargo la clase NEU se desempeño mejor que el gold standard, posiblemente debido a que la red que predice los aspectos es muy buena en inferir Propio_Partido por encima de los demás, ya que es la etiqueta que más aparece en el corpus, luego como la mayoría de textos que contienen este aspecto tienen como polaridad NEU la red que calcula la polaridad tiende a predecir esa etiqueta por encima del resto.

Capítulo 6

Conclusiones

En este trabajo se estudian técnicas para el reconocimiento de opiniones en medios sociales. En primer lugar podemos afirmar que construimos un sistema de aprendizaje automático que permite reconocer entidades y aspectos sobre los textos de los documentos para luego realizar análisis de sentimientos basado en los aspectos inferidos. Esto nos permite abordar el problema de entender las opiniones de los usuarios desde un punto de vista más específico que el cálculo de polaridades en un sistema de análisis de sentimientos convencional.

A lo largo del proyecto nos fuimos dando cuenta de que el problema de reconocimiento de entidades y aspectos en los textos es más desafiante de lo que a priori parece, ya que para diseñar una solución tuvimos que comprender cómo funcionan los algoritmos de clasificación multietiqueta y cómo estos nos permiten reconocer los aspectos sobre las entidades que un usuario expresa en el texto.

Sin lugar a dudas, creemos que se podrían haber obtenido mejores resultados si hubiéramos tenido un conjunto de entrenamiento etiquetado de opiniones más grande. En promedio, los corpus evaluados en este trabajo no superan las dos mil quinientas opiniones, cantidad muy pequeña para la complejidad del problema que se intenta resolver. Además, muchos de ellos están etiquetados con ejemplos de forma desbalanceada, como se pudo apreciar en la evaluación de resultados.

Sobre el sistema de aprendizaje, consideramos que existen partes del sistema que se pudieron haber optimizado, por ejemplo, el pre-procesamiento de los textos de los documentos, el ajuste de los parámetros de la red, o inclusive, probar con otra arquitectura de red. La contracara de esto fue que al tener corpus demasiado pequeños, no se pudo estudiar el desempeño del algoritmo frente al uso de distintos modelos de vectores de palabras.

Finalmente, por nuestra parte, cumplimos con éxito los objetivos propuestos. Sin embargo, sabemos que existen muchas actividades para realizar en este campo. Un ejemplo de esto es la extracción de expresiones temporales para las opiniones, actividad que fue descartada del alcance del proyecto al no obtener corpus de opiniones que contengan estos datos. Creemos que este proyecto reúne la teoría necesaria para abrir las puertas a futuros proyectos que requieran estudiar el análisis de sentimiento en un nivel específico y desarrollar herramientas para la recuperación de opiniones en los textos.

Trabajo a futuro

Como trabajo a futuro, además de las mejoras ya planteadas, se propone la construcción de un corpus de opiniones en español, suficientemente grande, con la estructura definida en este trabajo para luego utilizar el sistema de aprendizaje propuesto y evaluar las métricas con distintos modelos de vectores de palabras, a modo de comprender si un modelo en particular se desempeña mejor para este tipo de actividades.

Otra línea de trabajo paralela consiste en mejorar el sistema de aprendizaje implementado. Dentro de las posibles mejoras se incluye perfeccionar la lista de stop-words, para que no elimine palabras que pueden llegar a tener información de contexto; reemplazar emojis por palabras similares para agregar más información y mejorar el ruido en los textos mediante técnicas de stemming y lemmatization. Otra mejora podría ser evaluar otras arquitecturas de redes neuronales, tales como redes neuronales recurrentes, y comparar los resultados con los obtenidos en este trabajo.

Apéndice A

Descripción de los recursos utilizados

En este apéndice vamos a desarrollar la semántica de las etiquetas de los corpus vistos en el capítulo 2.4. Pretendemos simplemente que este apartado permita al lector entender un poco más el significado de las etiquetas propuestas en los corpus de las competencias SemEval y TASS.

A.0.1. Restaurant Reviews

Como vimos, Restaurant Reviews es un corpus de reseñas de Restaurantes de la SemEval en su edición 2015 para la tarea número 12 conocida como Aspect Based Sentiment Analysis (ABSA) [47]. Las entidades son representadas mediante anotaciones (etiquetas) de la forma $\{E\#A, OTE, P\}$ asignadas a nivel de oración en las reseñas teniendo en cuenta el contexto de la misma. A continuación vamos a explicar la semántica de las etiquetas asociadas:

Etiquetas para la entidad (Entity Labels)

Una entidad E de un par $\{E\#A\}$ se asocia a las siguientes etiquetas:

- **FOOD.** Para opiniones enfocadas en las comidas en general o en términos específicos, por ejemplo: Platos, opciones de almuerzos, desayunos, meriendas o cenas, etc.. Algunos ejemplos son:
 - Our food was great too! $\rightarrow \{FOOD\}$
 - The menu is very limited $\rightarrow \{FOOD\}$
 - The food was pretty traditional but it was hot and good with large portions. $\rightarrow \{FOOD\}$
 - The pizza was great. $\rightarrow \{FOOD\}$
- **DRINKS.** Para opiniones enfocadas en bebidas en general o en términos específicos del uso de la misma, por ejemplo: Bebidas, Gaseosas, Tragos, etc.. Algunos ejemplos son los siguientes:
 - Wonderful strawberry daiquiris as well! $\rightarrow \{DRINKS\}$
 - The food is good, especially their more basic dishes, and the drinks are delicious. $\rightarrow \{FOOD\}\{DRINKS\}$
 - Not enough wines by the glass either. $\rightarrow \{DRINKS\}$
 - The cream cheeses are out of this world and I love that coffee!! $\rightarrow \{FOOD\}\{DRINKS\}$

- SERVICE. Dedicado a opiniones enfocadas en el servicio en términos del cliente, la puntualidad del servicio, calidad del restaurante preparación de la comida, actitud del staff y su profesionalismo, el tiempo de espera, las opciones de envío ofrecida, etc.. A continuación, algunos ejemplos son:
 - The food here is rather good, but only if you like to wait for it. → {FOOD}{SERVICE}
 - The hostess is rude to the point of being offensive. → {SERVICE}
 - The takeout is great too since they give high quality tupperware as well. → {SERVICE}
 - I had a huge group for my birthday and we were well taken care of. → {SERVICE}
- AMBIENCE. Opiniones enfocadas en el ambiente del restaurante, su espacio interior o exterior, su decoración, opciones de entretenimiento, etc.. Los siguientes son ejemplos:
 - Very cozy and warm inside... → {AMBIENCE}
 - I highly recommend Cafe St. Bart's for their food, the ambience and wonderful service. → {FOOD}{AMBIENCE}{SERVICE}
 - The dining room is quietly elegant with no music to shout over, how refreshing! → {AMBIENCE}
 - Our family never expected such incredible entertainment in a restaurant. → {AMBIENCE}
- LOCATION. Aquellas enfocadas en el lugar del restaurante en términos de su lugar, la vista, el barrio, etc.. Ejemplos:
 - Conveniently located too, being right on Bedford ave. → {LOCATION}
 - The view is spectacular, and the food is great. → {LOCATION}{FOOD}
- RESTAURANT. Opiniones enfocadas al restaurante como un todo sin enfoque alguno de las siguientes entidades. Los siguientes son ejemplos:
 - It's a little out of the way if you don't live in the neighborhood, but definitely worth the trip from wherever you are. → {LOCATION}{RESTAURANT}
 - Please take my advice, go and try this place. → {RESTAURANT}

Cualquier otra opinión que no pueda ser expresada con las entidades descritas anteriores se considera fuera del alcance y se etiqueta de con *OutOfScope*. Entonces por ejemplo:

- I was in love with Pongsri on 48th, but compared to Suan it is slow in service and overpriced. → {OutOfScope}
- It's also attached to Angel's Share, which is a cool, more romantic bar... → {OutOfScope}
- The one on the east side is much better than the one on the west side. → {OutOfScope}

Etiquetas para los atributos (Attribute Labels)

Una atributo A de un par {E#A} se asocia a las siguientes etiquetas en el corpus:

- **GENERAL.** Etiqueta asignada a opiniones generales que expresan una opinión positiva o negativa (eventualmente neutral) sobre una entidad. Algunos ejemplos son:
 - My husband and I thought it would be great to go to the Jekyll and Hyde Pub for our anniversary, and to our surprise it was fantastic. → {RESTAURANT#GENERAL}
 - I received prompt service with a smile. → {SERVICE#GENERAL}
 - LOVE the atmosphere - felt like I was in Paris. → {AMBIENCE#GENERAL}
 - The location is perfect. → {LOCATION#GENERAL}
- **PRICES.** Asignado a opiniones de precios sobre entidades que hablan de comida, tragos o el restaurante como un todo. Ejemplos:
 - But the pizza is way too expensive. → {FOOD#PRICES}
 - Drinks way overpriced. → {DRINKS#PRICES}
 - The prices are wonderfully low. → {RESTAURANT#PRICES}
- **QUALITY.** Opiniones centradas en el gusto, la textura, la consistencia, la temperatura, la preparación, la cocción o calidad general de la comida y las bebidas servidas en el restaurante o la calidad del mismo como un todo. Estos son ejemplos de opiniones donde aparece:
 - Salads are a delicious way to begin the meal. → {FOOD#QUALITY}
 - Their bagels are fine, but they are a little overcooked, and not really a 'special' bagel experience. → {FOOD#QUALITY}
 - The drinks are amazing and half off till 8pm. → {DRINKS#QUALITY}, {DRINKS#PRICES}
- **STYLE&OPTIONS.** Opiniones que refieren a la presentación, estilo del servicio, tamaño de las porciones la variedad de la comida o el menú (platos innovadores, opciones vegetarianas o celíacos, etc..) y las bebidas servidas en el restaurante. Los siguientes son ejemplos:
 - The menu is very limited - I think we counted 4 or 5 entrees. → {FOOD#STYLEOPTIONS}
 - The portions are small but being that the food was so good makes up for that. → {FOOD#QUALITY}, {FOOD#STYLEOPTIONS}
 - Not enough wines by the glass either. → {DRINKS#STYLEOPTIONS}
- **MISCELLANEOUS.** Atributos que no entran en las las categorías mencionadas anteriormente. A continuación, los siguientes ejemplos:
 - Not a great place for family or general dining. → {RESTAURANT#MISCELLANEOUS}
 - Open late (well as late as I ever got there and I'm a night person) → {RESTAURANT#MISCELLANEOUS}

Expresión objetivo de la opinión (Opinion Target Expression, OTE)

Es una referencia explícita (mención en el texto) de la entidad E evaluada. Esta mención puede ser el nombre de la entidad, un nombre común o una secuencia de palabras de la misma. Algunos ejemplos son los siguientes:

- Leon is an East Village gem. → {RESTAURANT#GENERAL, Leon, positive}

- The food was bland oily. → {FOOD#GENERAL, food, negative}
- Great friendly service, Fast seating, Fast Delivery, Excellent sushi.
→ {SERVICE#GENERAL, service, positive}, {SERVICE#GENERAL, seating, positive},
{SERVICE#GENERAL, Delivery, positive}, {FOOD#GENERAL, sushi, positive}
- We love the food, drinks, and atmosphere.
→ {FOOD#QUALITY, food, positive},
{DRINKS#GENERAL, drinks, positive}, {AMBIENCE#GENERAL, atmosphere, positive}

Observaciones Toda oración que contiene más de una mención (por ejemplo, nominal y pronominal) a una misma entidad es anotada con aquella que sea más informativa.

- It's a nice place to relax and have conversation.
→ {AMBIENCE#GENERAL, place, positive}

Si una OTE tiene más de una ocurrencia en una misma entidad, solamente la primera debe ser anotada.

- Yes, they use fancy ingredients, but even fancy ingredients don't make for good pizza unless someone knows how to get the crust right. → {FOOD#QUALITY, ingredients, positive}

Toda entidad E implícitamente referenciada (ej. mediante pronombres) o inferida en la oración, su OTE es asignado con el valor "NULL", entonces por ejemplo:

- Everything was wonderful; → {RESTAURANT#GENERAL, NULL, positive}
- Conveniently located too, being right on Bedford ave.
→ {LOCATION#GENERAL, NULL, positive}
- I've waited over one hour for food. → {SERVICE#GENERAL, NULL, negative}
- Prices are in line. → {RESTAURANT#PRICES, NULL, positive}

Entonces, dado una reseña de un restaurante, se debe realizar la siguientes tareas:

- **Categorizar Aspectos** (Aspect Category): Identificar la entidad E y el atributo A asociado al par E#A de la opinión expresada.
- **Identificar la expresión objetivo de la opinión** (OTE): Esta referencia es identificada de forma única a través de anotar el lugar de comienzo y fin en el texto.
- **Polaridad de la opinión**: Cada par E#A de una oración debe ser asignado a un elemento del conjunto $P = \{positive, negative, neutral\}$ que representa la orientación del sentimiento.

Combinación de entidades y atributos en el par E#A: La siguiente tabla indica cómo se deben corresponder las entidades y los atributos de una opinión dada,

Categories Attributes	GENERAL	PRICES	QUALITY	STYLE&OPTIONS	MISCELLANEOUS
RESTAURANT	✓	✓	✗	✗	✓
FOOD	✗	✓	✓	✓	✗
DRINKS	✗	✓	✓	✓	✗
AMBIENCE	✓	✗	✗	✗	✗
SERVICE	✓	✗	✗	✗	✗
LOCATION	✓	✗	✗	✗	✗

A.0.2. Social-TV Corpus

Social-TV es un corpus presentado en la TASS de la SEPLN, el mismo es un corpus anotado a nivel de aspecto, donde en la página de la SEPLN la lista de aspectos definidos es la siguiente:

- Afición
- Árbitro
- Autoridades
- Entrenador
- Equipos: Equipo-Atlético_de_Madrid, Equipo-Barcelona, Equipo-Real_Madrid, Equipo (cualquier otro equipo)
- Jugadores: Jugador-Alexis_Sánchez, Jugador-Alvaro_Arbeloa, Jugador-Andrés_Iniesta, Jugador-Angel_Di_María, Jugador-Asier_Illarramendi, Jugador-Carles_Puyol, Jugador-Cesc_Fábregas, Jugador-Cristiano_Ronaldo, Jugador-Dani_Alves, Jugador-Dani_Carvajal, Jugador-Fábio_Coentrão, Jugador-Gareth_Bale, Jugador-Iker_Casillas, Jugador-Isco, Jugador-Javier_Mascherano, Jugador-Jesé_Rodríguez, Jugador-José_Manuel_Pinto, Jugador-Karim_Benzema, Jugador-Lionel_Messi, Jugador-Luka_Modric, Jugador-Marc_Bartra, Jugador-Neymar_Jr., Jugador-Pedro_Rodríguez, Jugador-Pepe, Jugador-Sergio_Busquets, Jugador-Sergio_Ramos, Jugador-Xabi_Alonso, Jugador-Xavi_Hernández, Jugador (cualquier otro jugador)
- Partido
- Retransmisión

La polaridad de los sentimientos fue anotada desde el punto de vista de un usuario de Twitter usando 3 etiquetas: P, NEU y N. No se realizó distinción alguna sobre si el autor del tweet no expresa sentimiento o expresa un sentimiento no positivo o no negativo.

A.0.3. STOMPOL

STOMPOL es un corpus de tweets en español para el análisis de sentimientos a nivel de aspectos basado en la campaña política del 2015 para las elecciones locales en España.

En la página de la SEPLN la lista de aspectos definidos es la siguiente:

- Economía: Impuestos, Infraestructura, Mercado, Políticas laborales...
- Sanidad: Hospitales, Sistema de servicio de salud público/privado, Drogas, Doctores...
- Educación: Estado de las escuelas publico/privadas, escolaridades...
- Propio_partido: Programa electoral, discursos, corrupción, críticas
- Otros_aspectos

Cada aspecto está relacionado a una o más entidades (separados por el símbolo pipe i.e " | " si un mismo aspecto habla de más de una entidad) que se corresponden con los principales partidos políticos de España:

- Partido_Popular (PP)
- Partido_Socialista_Obrero_Español (PSOE)
- Izquierda_Unida (IU)
- Podemos
- Ciudadanos (Cs)
- Unión_Progreso_y_Democracia (UPyD)

Cada tweet en el corpus fue manualmente anotado por dos lingüistas distintos y un tercero en caso de desacuerdo con la polaridad del sentimiento a nivel de aspecto. La polaridad de los sentimientos fue anotada desde el punto de vista de un usuario de Twitter usando 3 etiquetas: P, NEU y N. No se realizó distinción alguna sobre si el autor del tweet no expresa sentimiento o expresa un sentimiento no positivo o no negativo.

Bibliografía

- [1] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov (April, 2017). Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 427, 431.
- [2] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline. , 2014.
- [3] Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. Proceedings of SIGIR'15, pages 959–962, 2015.
- [4] Andrew B. Goldberg, Jerry Zhu, and Stephen Wright. Dissimilarity in graph-based semi-supervised classification. In Artificial Intelligence and Statistics (AISTATS), 2007.
- [5] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Tim Finin, and Anupam Joshi. Modeling trust and influence in the blogosphere using link polarity. In Proceedings of the International Conference on Weblogs and Social Media (ICWSM), 2007. Short paper
- [6] B. Liu. Opinion mining. «In Encyclopedia of Database Systems», pages 1986–1990. 2009.
- [7] Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. The Journal of Machine Learning Research, 3, 1137–1155.
- [8] Bethard, Steven, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. «Automatic extraction of opinion propositions and their holders». In Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text. 2004.
- [9] Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. Pattern Recognition, 37(9), 1757–1771. <https://doi.org/10.1016/j.patcog.2004.03.009>.
- [10] Brun C. et al. (2014) XRCE: Hybrid Classification for Aspect-based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 838 - 842, Dublin, Ireland, August 23-24, 2014.
- [11] Brychcin T. et al. (2014) UWB: Machine Learning Approach to Aspect-Based Sentiment Analysis, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 817 - 822, Dublin, Ireland, August 23-24, 2014.
- [12] Choi, Yejin, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In

- Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005). 2005.
- [13] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, and Kuksa P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.
- [14] Cristian Cardellino: Spanish Billion Words Corpus and Embeddings (March 2016), <https://crscardellino.github.io/SBWCE/>
- [15] D. Kingma and J. Ba, «Adam: A method for stochastic optimization,» arXiv preprint arXiv:1412.6980, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [16] Daniel Hopkins and Gary King. Extracting systematic social science meaning from text. Manuscript available at <http://gking.harvard.edu/files/words.pdf>, 2007 version was the one most recently consulted, 2007.
- [17] Dave, Kushal, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of International Conference on World Wide Web (WWW2003)*. 2003.
- [18] De Clercq O. et al. (2015) LT3: Applying Hybrid Terminology Extraction to Aspect-Based Sentiment Analysis, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.
- [19] Dowty, David R., Robert E. Wall, and Stanley Peters. *Introduction to Montague semantics*. Vol. 11. 1981: Springer.
- [20] E. Riloff and J. Wiebe, «Learning extraction patterns for subjective expressions,» *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.
- [21] E. Riloff, J. Wiebe, and T. Wilson, «Learning subjective nouns using extraction pattern bootstrapping,» *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pp. 25– 37 32, 2003.
- [22] G. Lee, J. Jeong, S. Seo, C. Kim, P. Kang, Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. *Know.-Based Syst.*, 152(C), 2018, pp. 70-82.
- [23] García-Cumbreras, M. Á., Villena-Román, J., Cámara, E. M., Díaz-Galiano, M. C., Martín-Valdivia, M. T., López, L. A. U.: Overview of TASS 2016. In: *Proceedings of TASS 2016: Workshop on Sentiment Analysis at SEPLN colocated with 31st SEPLN Conference*. CEUR Workshop Proceedings, España, 13–21(2016)
- [24] Goldberg Y. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 2016
- [25] Hamdan H. et al. (2015) Lsislif: CRF and Logistic Regression for Opinion Target Extraction and Sentiment Polarity Analysis, *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* pp. 719 - 724, Denver, Colorado, June 4-5, 2015.

- [26] Hobbs, Jerry R. and Ellen Riloff. «Information Extraction, in in Handbook of Natural Language Processing», 2nd Edition, N. Indurkha and F.J. Damerau, Editors. 2010, Chapman & Hall/CRC Press.
- [27] Hu, M and Liu, B. Mining and Summarizing Customer Reviews. Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), 2004.
- [28] Hu, M. and B. Liu. Mining and summarizing customer reviews. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), 2004.
- [29] Ian Goodfellow and Yoshua Bengio and Aaron Courville (2016). Deep Learning. MIT Press. p. 331.
- [30] Jeffrey Pennington, Richard Socher y Christopher Manning. «Glove: Global Vectors for Word Representation». En: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014.
- [31] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In Proceedings of the Third IEEE International Conference on Data Mining, pages 427–434.
- [32] Jindal, Nitin and Bing Liu. Identifying comparative sentences in text documents. in Proceedings of ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR-2006). 2006a.
- [33] Jindal, Nitin and Bing Liu. Mining comparative sentences and relations. in Proceedings of National Conf. on Artificial Intelligence (AAAI-2006). 2006b
- [34] Joachims, Thorsten. Making large-Scale SVM Learning Practical, in Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, Editors. 1999, MIT press
- [35] K. Tateishi, Y. Ishiguro and T. Fukushima, «A reputation search engine that gathers people's opinions from the Internet,» Technical Report NL-144-11, Information Proc-essing Society of Japan, pp.75-82, 2001 (in Japanese).
- [36] Kim, Soo-Min and Eduard Hovy. Determining the sentiment of opinions. in Proceedings of International Conference on Computational Linguistics (COLING-2004). 2004.
- [37] Kiritchenko S. et al. (2014) NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 437 - 442, Dublin, Ireland, August 23-24, 2014.
- [38] Le, Q., Mikolov, T. Distributed representations of sentences and documents. In: Proceedings of The 31st International Conference on Machine Learning, pp. 1188–1196 (2014).
- [39] LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto. 330, 352

- [40] Liu, Bing. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. 2006 and 2011: Springer.
- [41] Liu, Bing, Mingqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. in *Proceedings of International Conference on World Wide Web (WWW-2005)*. 2005.
- [42] Liu, Bing, and Lei Zhang. «A survey of opinion mining and sentiment analysis.» *Mining Text Data*. Springer US, 2012. 415-463.
- [43] Liu, Bing. «Sentiment analysis and opinion mining.» *Synthesis Lectures on Human Language Technologies 5.1* (2012): 1-167.
- [44] Liu, Bing. «Sentiment analysis and subjectivity.» *Handbook of natural language processing 2* (2010): 627-666.
- [45] Liu, Bing. «Sentiment analysis: A multi-faceted problem.» *IEEE Intelligent Systems* 25.3 (2010): 76-80.
- [46] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud Maria Jimenez-Zafra, and Gulsen Eryigit. 2016. Semeval-2016 Task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 19–30. <http://www.aclweb.org/anthology/S16-1002>.
- [47] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 Task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, Denver, Colorado, pages 486–495. <http://www.aclweb.org/anthology/S15-2082>.
- [48] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014t Task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 27–35. <http://www.aclweb.org/anthology/S14-2004>.
- [49] Martínez-Cámara, E., Díaz-Galiano, M. C., García-Cumbreras, M. Á., García-Vega, M., Viñena-Román, J: Overview of TASS 2017. In: *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN (TASS 2017)*, CEUR Workshop Proceedings, España, 13–21(2017)
- [50] Mathías and Dina Wonsever, "Spanish word vectors from Wikipedia", *Proceedings of Language Resources and Evaluation*, Portoroz, Slovenia, 05/2016.
- [51] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 327–335, 2006.

- [52] McCulloch, W. S. and W. H. Pitts: 1943, 'A Logical Calculus of the Ideas Immanent in Nervous Activity', *Bulletin of Mathematical Biophysics* 7, 115–133. Reprinted in McCulloch 1964, pp. 16–39.
- [53] Medhat, W., A. Hassan, y H. Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, pages 1093–1113.
- [54] Michael Laver, Kenneth Benoit, and John Garry. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(2):311–331, 2003
- [55] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Weinberger (eds.) *Advances in Neural Information Processing Systems* 26, pp. 3111–3119. Curran Associates, Inc. (2013)
- [56] Miles Efron. Cultural orientation: Classifying subjective documents by co-citation [sic] analysis. In *Proceedings of the AAAI Fall Symposium on Style and Meaning in Language, Art, Music, and Design*, pages 41–48, 2004.
- [57] Mitchell, Tom. M. 1997. *Machine Learning*. New York: McGraw-Hill.
- [58] Mohit Bansal, Claire Cardie, and Lillian Lee. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2008. Poster paper.
- [59] Mooney, Raymond J. and Razvan Bunescu. «Mining knowledge from text using information extraction». *ACM SIGKDD Explorations Newsletter*, 2005. 7(1): p. 3-10.
- [60] Pak, A., Paroubek, P. (2011) Classification en polarité de sentiments avec une représentation textuelle à base de sous-graphes d'arbres de dépendances. *TALN*
- [61] Pang, B., and Lee, L. (2008), «Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*» 2(1-2):1–135.
- [62] Pang, Bo and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of Meeting of the Association for Computational Linguistics (ACL-2004)*. 2004.
- [63] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of conference on empirical methods in natural Language Processing (EMNLP-2002)*. 2002.
- [64] Paredes-Valverde, M. A., Colomo-Palacios, R., SalasZárate, M. D. P., & Valencia-García, R. (2017). *Sentiment Analysis in Spanish for Improvement of Products and Services: A Deep Learning Approach*. Scientific Programming, 2017.
- [65] Paula Chesley, Bruce Vincent, Li Xu, and Rohini Srihari. Using verbs and adjectives to automatically classify blog sentiment. In *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, pages 27–29, 2006.
- [66] Peter Turney, «Thumbs up or thumbs down?», in *Semantic orientation applied to unsupervised classification of reviews*, pages 417424, 2002.

- [67] Pickett-Baker, J & Ozaki, R 2008, «Pro-environmental products: marketing influence on consumer purchase decision», *Journal of Consumer Marketing*, vol. 25, no. 5, p. 281.
- [68] Pierpaolo Basile and Valerio Basile and Danilo Croce and Marco Polignano. 2018. Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (AB-SITA). Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), CEUR.org, Turin
- [69] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3), 333–359. <https://doi.org/10.1007/s10994-011-5256-5>.
- [70] Rokach, L. (2010). "Ensemble-based classifiers". *Artificial Intelligence Review*. 33 (1–2): 1–39
- [71] Rosen-Zvi, Michal and Biehl, Michael and Kanter, Ido (1998). Learnability of periodic activation functions : General results. *Physical Review E*, 58(3), 3606.
- [72] Rosenblatt, Frank (1958), *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, Cornell Aeronautical Laboratory, *Psychological Review*, v65, No. 6, pp. 386–408
- [73] Rumelhart, D. E., Hinton, G. E., and Williams, G. E. (1986). Learning internal representations by error propagation, in «Parallel Distributed Processing» (D. E. Rumelhart and J. L. McClelland, Eds.), Vol. 1, MIT Press, Cambridge, MA.
- [74] S.ChandraKala and C. Sindhu, (2012), *Opinion Mining And Sentiment Classification: A Survey*, *ICTACT Journal on Soft Computing*, Vol- 03, ISSUE: 01, ISSN: 2229-6956
- [75] Sarawagi, Sunita. «Information extraction». *Foundations and Trends in Databases*, 2008. 1(3): p. 261-377.
- [76] Shawe-Taylor, John and Nello Cristianini. *Support Vector Machines*, 2000, Cambridge University Press.
- [77] Stephan Greene. *Spin: Lexical Semantics, Transitivity, and the Identification of Implicit Sentiment*. PhD thesis, University of Maryland, 2007.
- [78] Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Steede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 2011. 37(2): p. 267-307.
- [79] Tang, D., Qin, B., & Liu, T. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1422–1432.
- [80] Thet, T. T. & Na J. C. & Khoo C.S.G (2010) Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science* archive Volume 36 Issue 6 pp. 823-848 Sage Publications, Inc. Thousand Oaks, CA, USA.
- [81] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: neural networks for machine learning*, 4(2):26–31, 2012.

- [82] Toh Z. & Su J. (2015) NLANGP: Supervised Machine Learning System for Aspect Category Classification and Opinion Target Extraction, Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) pp. 719 - 724, Denver, Colorado, June 4-5, 2015.
- [83] Toh Z. & Wang W. (2014) DLIREC: Aspect Term Extraction and Term Polarity Classification System, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 235 - 240, Dublin, Ireland, August 23-24, 2014.
- [84] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 01 2013.
- [85] Tony Mullen and Robert Malouf. A preliminary investigation into sentiment analysis of informal political discourse. In AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI/CAAW), pages 159–162, 2006.
- [86] V. Hatzivassiloglou and K. R. McKeown, «Predicting the Semantic Orientation of Adjectives,» in Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL 1997). Morgan Kaufmann Publishers / ACL, 1997, pp. 174–181.
- [87] Villena-Román, Julio, Janine García-Morera, Miguel A., García Cumberras, Eugenio Martínez Cámara, M.Teresa Martín Valdivia, and L. Alfonso Ureña López, (eds). 2015. Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN. CEUR WS Vol 1397. Disponible en: <http://ceur-ws.org/Vol-1397>
- [88] Wagner J. et al. (2014) DCU: Aspect-based Polarity Classification for SemEval Task 4, Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) pp. 223 - 229, Dublin, Ireland, August 23-24, 2014.
- [89] Werbos, P. (1974) «Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences,» Harvard University dissertation
- [90] Wiebe, Janyce, Rebecca F. Bruce, and Thomas P. O'Hara. Development and use of a gold-standard data set for subjectivity classifications. in Proceedings of the Association for Computational Linguistics (ACL-1999). 1999.
- [91] Wiebe, Janyce, Theresa Wilson, and Claire Cardie. «Annotating expressions of opinions and emotions in language». Language Resources and Evaluation, 2005. 39(2): p. 165-210.
- [92] Wiebe, Janyce. Identifying subjective characters in narrative. in Proceedings of the International Conference on Computational Linguistics (COLING-1990). 1990.
- [93] Xianghua, F., L. Guo, G. Yanyan, y W. Zhiqiang. 2013. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon.
- [94] Y. Kim, Convolutional neural networks for sentence classification, in: Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

- [95] Yam-Tang, E.P.Y. and Chan, R.Y.K. (1998). «Purchasing behaviours and perceptions of environmentally harmful products», *Marketing Intelligence & Planning*, 16 (6), 356-362.
- [96] Zhiqiang Toh and Jian Su. 2016. Nlangp at semeval 2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 282–288.