

# **Clasificación de aspectos en opiniones textuales con técnicas de aprendizaje profundo y continuo**



**Dionis López Ramos**

Centro de Investigaciones de la Informática

Universidad Central "Marta Abreu" de Las Villas

Tesis presentada en opción al grado científico de

*Doctor en Ciencias Técnicas*

Santa Clara, 2023

# **Clasificación de aspectos en opiniones textuales con técnicas de aprendizaje profundo y continuo**



**Dionis López Ramos**

Centro de Investigaciones de la Informática

Universidad Central “Marta Abreu” de Las Villas

Tesis presentada en opción al grado científico de

*Doctor en Ciencias Técnicas*

Tutor: Dr.C. Fernando José Artigas Fuentes

Santa Clara, 2023

Me gustaría dedicar este trabajo a mis queridos padres y a todos los profesionales, amigos y familiares que han incidido en mi formación como investigador.

## **AGRADECIMIENTOS**

Me gustaría agradecer:

- A Dios, por estar siempre conmigo, por alumbrar el camino, por las enseñanzas y la fuerza.
- A mi mamá y a mi papá, por las enseñanzas de siempre, por reír, por llorar junto a mí, por el ejemplo.
- A mi país y a la Revolución, por la oportunidad de crecer y ser mejor profesional.
- Al Dr.C. Rafael Bello y a muchos profesionales excelentes del Centro de Investigaciones en Informática de la Universidad Central “Marta Abreu” de Las Villas, por la maravillosa oportunidad de formarme como investigador y como mejor persona.
- A mi tutor, Dr.C. Fernando Artigas Fuentes.
- Al Dr.C. Antolín Romero, que ha sido mentor y apoyo en estos años.
- Al Dr.C. Erislandy Omar, al Dr.C. Sergio Cano y al Ms.C. Oscar Au, de la Universidad de Oriente, quienes me apoyaron y motivaron en todo momento.
- Al Dr.C. José Manuel Badia Contellez, de la Universidad Jaume I de Castellón (Comunidad Valenciana, España), por su apoyo en esta investigación.
- A mis amigos Alejandro Bergues y Roberto Acosta, por su mecenazgo y apoyo en esta investigación, parte importante de los resultados.
- A mis familiares, que siempre han creído en mí y en que llegaría al éxito (*Veni, vidi, vici*).

## SÍNTESIS

La clasificación de sentimientos basada en aspectos de opiniones textuales es una tarea muy importante dentro del análisis de sentimientos o minería de opiniones. La gran cantidad de datos obtenidos en este proceso logra mayor exactitud al analizar la información y contribuye a la toma de decisiones. Varios modelos de aprendizaje profundo han obtenido resultados relevantes en el análisis de sentimientos basado en aspectos en dominios específicos, como opiniones sobre restaurantes u hoteles. Sin embargo, cuando estos modelos son empleados de forma continua en el aprendizaje de múltiples dominios decrece significativamente la calidad de los resultados y aparecen problemas, como el olvido catastrófico del conocimiento aprendido en dominios anteriores.

En esta investigación se propone un nuevo modelo, combinando un método de aprendizaje continuo y otro de aprendizaje profundo, para ser aplicado en el análisis de sentimientos basado en aspectos, que muestra resultados comparables o superiores a otros algoritmos reportados en la literatura. Los resultados son evaluados empleando medidas que tienen en cuenta el desbalance de los datos y el aprendizaje continuo. Como resultado práctico, se introduce un módulo para el análisis de sentimientos en inglés, basado en el modelo diseñado en esta investigación en una plataforma para la gestión digital de campañas de bien público.

## **ABSTRACT**

Sentiment classification based on textual opinion aspects is a very important task within Sentiment Analysis or Opinion Mining. The large data obtained in this process achieves greater accuracy when analyzing the information and contributes to decision making. Several Deep Learning models have obtained relevant results in sentiment analysis based on aspects closed to specific domains such as opinions about restaurants or hotels. However, when these models are used in Continual Learning scenarios with multiple domains, the effectiveness of the results decreases significantly, and problems such as the catastrophic forgetting of the knowledge learned in previous domains appear.

A new model combining a Continual and Deep Learning method to be applied in aspect-based sentiment analysis, which shows comparable or superior results to other models reported in the study, is proposed. The results are evaluated using measures that take into account data imbalance and continual learning. A module for sentiment analysis in English, which is a platform for the digital management of public wellness campaigns based on the model designed, is introduced as a practical result in this research.

# ÍNDICE

<b>FIGURAS</b>	<b>IX</b>
<b>TABLAS</b>	<b>XI</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>1. Acerca del análisis de sentimientos basado en aspectos</b>	<b>8</b>
1.1. Características generales del análisis de sentimientos basado en aspectos . . . . .	9
1.2. Clasificación textual . . . . .	11
1.3. Formas de representación textual . . . . .	15
1.3.1. Vector de palabras embebidas ( <i>Word Embeddings</i> ) . . . . .	16
1.3.2. Representación de Codificador Bidireccional de Transformadores ( <i>Bi-directional Encoder Representation with Transformer</i> ; BERT) . . . . .	17
1.4. Extracción y clasificación de aspectos . . . . .	18
1.4.1. Redes Neuronales Convolucionales ( <i>Convolutional Neural Network</i> ; CNN) . . . . .	18
1.4.2. Redes Neuronales Recurrentes ( <i>Recurrent Neural Network</i> ; RNN) . .	19
1.4.3. Mecanismos de Atención . . . . .	21
1.4.4. Redes de Memoria (ReM) . . . . .	22
1.4.5. Grafos Convolucionales Neuronales (GCN) . . . . .	23
1.5. Propuestas de modelos de aprendizaje continuo . . . . .	24
1.5.1. Definición del problema y notación . . . . .	26
1.5.2. Clasificación de los modelos de aprendizaje continuo . . . . .	27
1.5.3. Modelos de regularización de aprendizaje continuo . . . . .	29

---

## ÍNDICE

1.6. Consideraciones finales del capítulo . . . . .	35
<b>2. Propuesta de modelos computacionales para el análisis de sentimientos basado en aspectos</b> <span style="float: right;"><b>36</b></span>	
2.1. Propuesta de modelo computacional para la extracción de aspectos . . . . .	36
2.1.1. Descripción del modelo para la extracción de aspectos en múltiples dominios . . . . .	37
2.1.2. Análisis de la complejidad computacional del entrenamiento del modelo para la extracción de aspectos . . . . .	47
2.2. Análisis experimental del método para la extracción de aspectos . . . . .	50
2.3. Propuesta de modelo computacional para la clasificación de aspectos . . . . .	55
2.3.1. BERT como capa de vectores embebidos . . . . .	56
2.3.2. Diseño del modelo base para la clasificación de aspectos . . . . .	57
2.3.3. Modelo para la clasificación de aspectos . . . . .	58
2.4. Medidas para evaluar la calidad del aprendizaje de los modelos . . . . .	63
2.4.1. Medidas para estimar el olvido catastrófico en modelos de aprendizaje continuo . . . . .	64
2.5. Conjuntos de datos empleados para el aprendizaje de los modelos . . . . .	66
2.5.1. Estudio de la cercanía semántica de los conjuntos de datos . . . . .	67
2.6. Evaluación de la propuesta para la clasificación de aspectos . . . . .	68
2.6.1. Detalles de implementación de los experimentos . . . . .	68
2.6.2. Descripción de las experimentaciones . . . . .	69
2.6.3. Modelos del estado del arte para la evaluación del aprendizaje profundo y continuo . . . . .	71

## ÍNDICE

---

2.6.4. Evaluación de la estrategia de comparación con otros modelos del estado del arte . . . . .	72
2.6.4.1. Prueba de Ablación . . . . .	74
2.6.4.2. Evaluación de la propuesta LLA y una propuesta reciente del estado del arte . . . . .	74
2.6.5. Análisis de la complejidad computacional del entrenamiento del modelo para la clasificación de aspectos . . . . .	78
2.6.6. Evaluación de la estrategia de influencia del orden del aprendizaje de los dominios . . . . .	82
2.7. Conclusiones parciales . . . . .	84
<b>3. El análisis de sentimientos basado en aspectos para la gestión de campañas de bien público</b>	<b>88</b>
3.1. La gestión de la información en las campañas de bien público . . . . .	88
3.1.1. Procesamiento de la información en las campañas de bien público: Herramientas digitales . . . . .	89
3.1.2. El análisis de sentimientos basado en aspectos para el procesamiento de la información . . . . .	90
3.2. WisePocket: Plataforma digital para campañas de bien público . . . . .	93
3.3. Módulo para el análisis de sentimientos en WisePocket . . . . .	95
3.3.1. Caso de uso: Campaña de bien público para la divulgación del XIII Seminario Nacional sobre Estudios Canadienses . . . . .	96
3.3.2. Otros escenarios en Cuba para el uso de los resultados de la investigación	97
3.3.3. Consideraciones necesarias para extender los modelos propuestos al idioma español . . . . .	98

## **ÍNDICE**

---

3.4. Conclusiones parciales . . . . .	100
<b>CONCLUSIONES</b>	<b>102</b>
<b>REFERENCIAS</b>	<b>104</b>
<b>ANEXOS</b>	<b>121</b>
<b>A. Producción científica del autor</b>	<b>122</b>
A.1. Publicaciones en revistas y conferencias . . . . .	122
A.1.1. Otras producciones científicas relacionadas con el investigador . . . . .	122
A.2. Participación en eventos internacionales y nacionales . . . . .	123
A.3. Registros de software . . . . .	123
<b>B. Principales características de los modelos de aprendizaje profundo</b>	<b>124</b>
<b>C. Diagrama del proceso de entrenamiento del modelo de extracción de aspectos</b>	<b>126</b>
<b>D. Diferencias entre aprendizaje continuo y otras estrategias</b>	<b>128</b>
<b>E. Módulos de la plataforma Wisepocket</b>	<b>133</b>
<b>F. Trabajos que combinan el aprendizaje continuo y el análisis de sentimientos</b>	<b>135</b>
<b>G. Representación textual</b>	<b>140</b>

# FIGURAS

1.1.	Taxonomía de subtareas para ABSA, tomado de [1] . . . . .	10
1.2.	Taxonomía de métodos de Aprendizaje Profundo para ABSA. . . . .	15
1.3.	Arquitectura de una CNN de siete capas. . . . .	19
1.4.	Arquitectura de una RNN de tres capas. . . . .	20
1.5.	Esquema general de modelos de Aprendizaje Continuo con estrategia de Regularización. . . . .	30
2.1.	Etapas para la creación de un modelo para la extracción de aspectos basado en el aprendizaje profundo y continuo. . . . .	37
2.2.	Esquema general del modelo para la extracción de aspectos Lwf-CNN-lgR. . .	39
2.3.	Resultados de la medida F1 con dominio cruzado. . . . .	52
2.4.	Resultados de la medida F1 en las pruebas en el dominio. . . . .	53
2.5.	Resultados del experimento sobre la reducción del olvido catastrófico. . . .	54
2.6.	Representación de la información de entrada y salida durante el proceso de aprendizaje (P: positivo, N: negativo, Neu: neutral). . . . .	57
2.7.	Similaridad coseno entre los centroides de cada dominio. . . . .	68
2.8.	Evaluación de los modelos con la prueba de Holm con un nivel de especificidad de 0.05 para la medida de F1-macro. . . . .	82
2.9.	Evaluación de los modelos con la prueba de Holm con un nivel de especificidad de 0.05 para la medida Kappa. . . . .	82
2.10.	Resultados de <i>BSpLLA</i> para los seis ordenamientos de los conjuntos de entrenamiento. . . . .	83
3.1.	Diagrama de despliegue donde se muestra el uso de los modelos <i>Learning without Forgetting with Linguistic Rules</i> ; Lwf-CNN-lgR) y ( <i>Lifelong Learning of Aspects</i> ; LLA) en la plataforma WisePocket para la retroalimentación de las opiniones de una campaña de bien público. . . . .	94
C.1.	Diagrama del flujo del entrenamiento del modelo Lwf-CNN-lgR. . . . .	127
E.1.	Módulos de la plataforma WisePocket. . . . .	134

## FIGURAS

# TABLAS

2.1.	Resultados en dominio cruzado para opiniones de restaurantes y hoteles. . . . .	53
2.2.	Descripción de los conjuntos de datos usados. . . . .	66
2.3.	Ejemplo del desbalance entre las clases en los conjuntos de datos. . . . .	67
2.4.	Acrónimos y nombres de los modelos comparados durante la evaluación de la nueva propuesta. . . . .	72
2.5.	Promedio de los resultados al emplear diferentes modelos base y la estrategia de AC propuesta en esta investigación. . . . .	72
2.6.	Promedio de los resultados entre <i>BSpLLA</i> y otras propuestas en SOTA. . . . .	73
2.7.	Promedio de los resultados de la ablación entre <i>BSpLLA</i> y <i>BSp</i> . . . . .	74
2.8.	Resultados de la propuesta de en [2](según artículo) y los de <i>BSpLLA</i> . . . . .	75
2.9.	Resultados de los experimentos para estimar el mejor desempeño entre [2] y <i>BSpLLA</i> . . . . .	77
2.10.	Valores de F1-macro del modelo propuesto para cada ordenamiento de los conjuntos de datos. . . . .	83
B.1.	Modelos de aprendizaje profundo usados en trabajos sobre ABSA. . . . .	125

## Glosario de términos

- **AP:** Aprendizaje Profundo.
- **AC:** Aprendizaje Continuo.
- **ABSA:** Siglas en inglés de Aspect Based Sentiment Analysis (Análisis de Sentimientos Basado en Aspectos).
- **PLN:** Procesamiento del Lenguaje Natural.
- **MA:** Mecanismos de Atención.
- **BERT:** Siglas en inglés de Bidirectional Encoder Representation with Transformer (Representación de Codificador Bidireccional de Transformadores).
- **TIL:** Siglas en inglés de Tasks Incremental Learning (Aprendizaje Incremental de Tareas).
- **DIL:** Siglas en inglés de Domains Incremental Learning (Aprendizaje Incremental de Dominios).
- **LSTM:** Siglas en inglés de Long Short Term Memory (Memoria de Corto Plazo).
- **GRU:** Siglas en inglés de Gated Recurrent Unit (Unidades Recurrentes con Compúteras).
- **BLSTM:** Siglas en inglés de Bidirectional Long Short Term Memory (Memoria de Corto Plazo Bidireccionales).
- **CNN:** Siglas en inglés de Convolutional Neural Network (Redes Convolucionales Neuronales).
- **SVM:** Siglas en inglés de Support Vector Machine (Máquinas de Vector de Soporte).
- **CRF:** Siglas en inglés de Conditional Random Field (Campos Condicionales Aleatorios).
- **BC:** Base de Conocimientos.

- **DG:** Descenso del Gradiente.
- **DGE:** Descenso del Gradiente Estocástico.
- **SI:** Siglas en inglés de Synaptic Intelligence (Inteligencia Sináptica).
- **LLA:** Siglas en inglés de Lifelong Learning of Aspects (Aprendizaje Continuo de Aspectos).
- **AR1:** Siglas en inglés de Architectural and Regularization 1 (Arquitectura y Regularización 1).
- **Lwf-CNN-IgR:** Siglas en inglés de Learning without Forgetting with Linguistic Rules (Aprendizaje sin olvido con redes neuronales convolucionales y con reglas lingüísticas)
- **Lwf-CNN:** Siglas en inglés de Learning without Forgetting with Convolutional Neural Network (Aprendizaje sin olvido con redes neuronales convolucionales)
- **EWC:** Siglas en inglés de Elastic Weight Consolidation (Consolidación elástica de pesos).
- **LLM:** Siglas en inglés de Lifelong Learning Memory (Memorias de Aprendizaje Continuo).
- **BSp:** Siglas en inglés BERT Special (Bert especial).
- $AE_{EwC}$ : Siglas en inglés Attentional Encoder Network con BERT.
- $AE_{LLA}$ : Attentional Encoder Network con BERT y el nuevo modelo LLA.
- $AE_{AR1}$ : Attentional Encoder Network con BERT y AR1.
- $BSp_{EwC}$ : BERT Special con EwC.
- $BSp_{LLA}$ : BERT Special con el nuevo modelo LLA.
- $AT_{LLA}$ : Attentional Encoder Network con LSTM y el nuevo modelo LLA.
- $LC_{LLA}$ : Local Context Focus con BERT y el nuevo modelo LLA.
- **Accr:** Exactitud, medida para evaluar el desempeño de los modelos computacionales.
- **Kappa:** Cohen-Kappa, medida para evaluar el desempeño de los modelos computacionales en conjuntos desbalanceados.

- **F1-macro:** Macro F1, medida para evaluar el desempeño de los modelos computacionales.
- **OvrcForgtt:** Medida para evaluar el olvido catastrófico en modelos computacionales.
- $w$ : Palabra presente en una oración o en un grupo de palabras.
- $n$ : Cantidad de muestras o ejemplos de un conjunto de datos (entrenamiento/validación/pruebas).
- $\theta$ : Conjunto de parámetros a entrenar en un modelo computacional.
- $m$ : Cantidad de conjuntos de entrenamiento para el aprendizaje de un modelo computacional de forma secuencial.
- $b$ : Cantidad de elementos de un lote (batch), en un conjunto de datos.
- $p$ : Cantidad de ciclos (iteraciones) sobre un conjunto de datos para el entenamiento de un modelo de Aprendizaje Profundo.
- $r$ : Tamaño de la secuencia o vector de entrada a cada capa de una red neuronal.
- $d$ : La dimensión del vector de entrada a una capa de una red neuronal.
- $k$ : Tamaño del kernel de las convoluciones de un modelo de redes neuronales convolucionales.
- **textbfc:** Cantidad de clases de clasificación durante el entrenamiento de un modelo computacional.

# INTRODUCCIÓN

La información procedente de diversas fuentes digitales crece constantemente. Una gran parte de esta son textos no estructurados que poseen un conocimiento valioso para diversas áreas de la sociedad [1, 2]. Una de las fuentes que producen información relevante son las opiniones sobre productos, eventos y personalidades [3, 4].

El estudio de las opiniones o análisis de sentimientos es importante en diversas áreas de la sociedad, tales como la salud, el gobierno y la economía [5]. En los últimos años han aumentado las herramientas computacionales para el análisis de sentimientos (p.ej., SentiStrength<sup>1</sup>, Repustate<sup>2</sup>, Lexalytics<sup>3</sup>). Compañías incipientes y grandes empresas como Microsoft, Google, Hewlett-Packard y Adobe tienen sus propuestas para el análisis de opiniones sobre sus productos y servicios [4]. Ejemplos del uso del análisis de sentimientos son: cómo fue usada negativamente la opinión pública en el escándalo de Cambridge Analytica-Facebook relacionado con las elecciones presidenciales de 2016 en Estados Unidos [6] y el análisis de las opiniones de los clientes de hoteles o alojamientos para mejorar sus servicios [7].

El Procesamiento del Lenguaje Natural (*Natural Language Processing*; NLP) reúne varias herramientas computacionales y conceptos de la Lingüística Computacional, importantes para comprender y procesar datos no estructurados asociados a las opiniones en textos digitales [4, 8, 9]. Una de las tareas del NLP es el análisis de sentimientos o la minería de opiniones. Esta tarea se centra en el estudio computacional de las opiniones, evaluaciones, actitudes y emociones que expresan las personas acerca de productos, servicios, organizaciones, individuos y eventos [4, 8].

Desde el comienzo del siglo XXI el análisis de sentimientos o minería de opiniones ha sido un área de investigación muy activa dentro del campo del NLP [5, 8, 10]. Sin embargo, la naturaleza subjetiva y heterogénea de la información y los datos no estructurados hacen que todavía existan retos en esta área [8]. Las propuestas existentes tienen un alcance a nivel de documentos, oraciones o aspectos (características). Aunque el análisis a nivel de documento y

---

<sup>1</sup><http://sentistrength.wlv.ac.uk/>

<sup>2</sup><https://www.repustate.com/>

<sup>3</sup><https://www.lexalytics.com/>

de oración es muy ventajoso, para realizar un análisis a mayor profundidad es más útil el nivel de “aspectos”. Esto se debe a que el análisis de sentimientos a nivel de aspectos se encarga de identificar “aspectos” asociados a “entidades” (p.ej., nombres de productos, servicios, personas, organizaciones y eventos) en un fragmento del texto analizado (p.ej., frase, oración o párrafo) y posteriormente clasifica los sentimientos presentes en este conjunto de información (e.d., “entidades” y “aspectos”) [4, 11].

El análisis de sentimientos a nivel de aspectos es conocido en la literatura científica como Análisis de Sentimientos Basado en Aspectos (*Aspect Based Sentiment Analysis*; ABSA) [4]. Esta tarea fue llamada inicialmente análisis de sentimientos basado en características [4]. ABSA es una tarea compleja por el grado de subjetividad que puede tener el sentimiento presente en un aspecto con respecto a su entidad. En una oración o frase pueden aparecer aspectos y entidades distintas. Los aspectos pueden estar representados por diferentes palabras o sinónimos. El uso de modificadores de opinión en uno o varios aspectos puede variar el sentimiento expresado en ellos [4, 12]. Generalmente se clasifican los sentimientos de los aspectos en positivo, negativo o neutro [4]. Esta clasificación también tiene el reto de diferenciar el valor neutro del resto de los valores. Además de reconocer los aspectos, es necesario extraer la entidad a la cual están asociados [4, 8, 9].

Algunos trabajos [13, 14] han tratado de enfrentar esta tarea empleando reglas lingüísticas o diccionarios de palabras, pero la construcción de estos recursos es muy costosa por el tiempo en la creación de soluciones, especialistas necesarios, entre otros aseguramientos.

Una alternativa para lograr la extracción y clasificación de aspectos ha sido el empleo de diferentes modelos computacionales de aprendizaje automatizado sobre conjuntos de entrenamiento de dominios específicos (p.ej., opiniones sobre restaurantes o efectos electrodomésticos). La construcción de estos conjuntos puede tener un alto costo, que se incrementa cuando se desea extender a otros dominios o idiomas como el español [4, 8, 15]. Una de las posibles soluciones a este problema consiste en encontrar modelos que permitan aprender de varios conjuntos de datos y reconocer de forma automática reglas o patrones lingüísticos que puedan ser extensibles a otros dominios del conocimiento o idiomas [16].

Uno de los modelos computacionales que ha mejorado la efectividad en los resultados, al aplicarlo a varios tópicos del conocimiento humano (p.ej., procesamiento de imágenes, procesamiento del lenguaje natural, entre otros), es el aprendizaje profundo (AP), conocido en

la literatura por (*Deep Learning*; DL) [8, 17]. Las propuestas basadas en el AP tienen varias mejoras sobre los modelos de aprendizaje automatizado tradicionales (p.ej., Modelos Ocultos de Markov, Máquinas de Vector de Soporte, Modelos Latentes de Dirichlet). Dos de sus más destacados aportes consisten en permitir trabajar con datos en bruto, reduciendo la necesidad de realizar un elaborado pre-procesamiento de la información, y evitar la ingeniería de características en los conjuntos de entrenamiento [17].

Algunos enfoques que usan modelos de AP han sido usados en ABSA y han mejorando la efectividad de los resultados al compararlos con otros modelos tradicionales del aprendizaje automatizado [5, 8, 10, 18]. Los modelos de AP en ABSA permiten obtener de forma automática reglas o patrones lingüísticos [8]. Entre los modelos de este tipo empleados en ABSA se destacan, por su efectividad, los llamados Memoria a Corto Plazo (*Long Short Term Memory*; LSTM) [19] y las Unidades Recurrentes Cerradas (*Gated Recurrent Units*; GRU) [20] y sus variantes Bidireccionales [18].

Los modelos computacionales anteriormente referenciados son empleados con gran efectividad en problemas donde existe una secuencia o serie de eventos en el tiempo [8]. La naturaleza secuencial del análisis de textos digitales o palabras en una oración ha motivado a varios investigadores a usarlos para el ABSA [1].

Otros enfoques de AP emplean los Mecanismos de Atención [21, 22] y los Grafos Convolucionales Neuronales [23], que permiten conservar la relación semántica entre los aspectos y las palabras del contexto y obtener mejores resultados que las propuestas que usan solamente LSTM o GRU [5, 8].

La forma de representación textual está asociada a la organización de la información para poder ser empleada por diferentes modelos computacionales y sistemas de información [24]. La eficacia de la forma de representación influye en el entrenamiento de los modelos de AP, debido a que se basan esencialmente en el trabajo con redes neuronales [17]. Uno de los retos más importantes al utilizar redes neuronales es lograr una forma de representación correcta para los datos de entrada a la red. En ABSA, los conjuntos de datos para el entrenamiento de estas redes están formados por documentos u oraciones [25].

Una de las formas de representación de la información textual para el NLP basado en redes neuronales son las Palabras Embebidas (*Word Embeddings*) [26]. Este es el nombre de un conjunto de modelos de lenguaje y técnicas de aprendizaje en el que las palabras o frases

del vocabulario se vinculan a vectores de números reales. *Word Embeddings* conceptualmente transforma un espacio con una dimensión por cada palabra a un espacio vectorial continuo con menos dimensiones. Algunos modelos de *Word Embeddings* usados en diferentes trabajos son Skip-Gram y CBOW [26], Glove [27] y Fasttext [28]. Cada uno presenta diferencias en la forma en que fueron creados los vectores de palabras a partir de los conjuntos de datos de entrenamiento. Para cada uno de ellos se proponen modelos pre-entrenados<sup>4</sup> con grandes conjuntos de datos como Wikipedia en inglés, o permiten que se construyan mediante el uso de herramientas computacionales como Word2vec<sup>5</sup>.

Otros modelos de lenguajes, y sus propuestas pre-entrenadas, han sido empleados para representar computacionalmente la semántica presente en los textos. Algunos ejemplos son OpenIA GTP [29], ELMo [30] y la Representación de Codificador Bidireccional de Transformadores (*Bidirectional Encoder Representations from Transformers*; BERT) [31]. Por otro lado, el empleo de BERT en la tarea ABSA permite obtener mejores valores de calidad con respecto a otras propuestas del estado del arte [31, 32, 33]. Este se ha convertido en una de las más exitosas formas de representación con respecto a otros *Word Embeddings* en tareas de NLP [8].

A pesar de los buenos resultados obtenidos por los modelos de AP en ABSA, por lo general, estos realizan la extracción y clasificación de aspectos en un solo dominio del conocimiento [34, 35]. Cuando estas propuestas son aplicadas a diferentes dominios su efectividad disminuye [8]. Esta limitación, asociada a la cantidad de dominios durante el aprendizaje de la red neuronal, reduce el alcance de estos modelos.

Uno de los conceptos que permite el uso de un único modelo para resolver la clasificación o agrupamiento de información, en varias tareas o dominios de forma secuencial es el Aprendizaje Continuo (AC) (también es conocido en la literatura como Aprendizaje Permanente (*Lifelong Learning*) o Aprendizaje Incremental (*Incremental Learning*)) [16, 36]. Este tipo de aprendizaje ha representado un reto para los modelos de aprendizaje en general, y las redes neuronales en particular, al desarrollar sistemas de Inteligencia Artificial para diversos problemas o interrogantes asociados al diseño de estos modelos [36, 37, 38].

Uno de los problemas o retos a los que deben enfrentarse los modelos que aplican el AC consiste en conservar, durante el proceso de aprendizaje de varias tareas o dominios, la efectividad para identificar los patrones o características comunes (p.ej., en el caso de la tarea ABSA el as-

---

<sup>4</sup><https://nlp.stanford.edu/projects/glove/>

<sup>5</sup><https://code.google.com/archive/p/word2vec/>

pecto “precio” es común para dominios de opiniones sobre restaurantes, hoteles y dispositivos electrónicos) [16]. La efectividad de estos modelos es útil para la tarea ABSA, porque permite reducir los costos de aprendizaje y de cómputo [8].

Para una definición formal del AC con redes neuronales (p.ej., AP), podemos establecer que dada una secuencia de conjuntos de datos de entrenamiento  $D_1, D_2, \dots, D_T$  uno a la vez, el objetivo es entrenar el modelo  $f_T : X \rightarrow Y$  en la tarea  $T$  (e.d., tarea o dominio del conocimiento), después de ser secuencialmente entrenado en los conjuntos de datos anteriores [36, 37]. En  $f_T$ , el valor de  $X$  es el espacio de entrada, representado en el NLP por un vector de valores reales y  $Y$  es el vector de inferencia de la clasificación que puede ser  $k$ -dimensional (e.d., Para ABSA  $k = 3$ : positivo, negativo, neutro).

A partir de su definición formal, un modelo de AC tiene como restricción en su configuración que durante el aprendizaje de la tarea  $T$  no es posible acceder a los datos de las tareas previas, pero puede tenerse una cantidad limitada de información acerca de las mismas. El modelo, al ser entrenado usando el conjunto de datos actual  $D_T$ , puede olvidar cómo predecir para las instancias del conjunto de datos  $D_t; t < T$ . Este problema es conocido como olvido catastrófico [16, 36, 37, 39] y ocurre cuando las redes neuronales son secuencialmente entrenadas en muchas tareas, debido a que los pesos de la red estimados para una tarea  $A$  son modificados durante el proceso de aprendizaje de la tarea  $B$  [40].

Algunas propuestas han tratado de superar o disminuir el olvido catastrófico, por ejemplo en la clasificación de imágenes [40] y los juegos de estrategia [41]. Sin embargo, no existen muchas propuestas para la tarea ABSA y estas no tienen los resultados de efectividad deseados [5, 8, 25]. El desarrollo de modelos de AC para la tarea ABSA constituye un reto para la investigación científica en el campo del NLP.

El análisis histórico-lógico en torno a la tarea ABSA y el uso de modelos de AP permiten determinar que aún persisten insuficiencias al clasificar aspectos a partir de conjuntos de opiniones de varios dominios del conocimiento, debido a la reducción de la efectividad durante este proceso. Esto constituye un problema al cual aún la ciencia no ha dado respuestas definitivas, lo cual justifica el planteamiento del siguiente **problema de investigación**: La baja efectividad de los métodos de clasificación cuando se aplican sobre colecciones de opiniones de múltiples dominios.

El **objetivo general** de esta investigación es lograr el diseño e implementación de **modelos de**

**aprendizaje computacional** eficaces y **eficientes** para la extracción y clasificación de aspectos, en opiniones de múltiples dominios **en idioma inglés**, basadas en modelos de aprendizaje profundo y continuo.

A partir de este objetivo general se derivan los siguientes objetivos específicos:

- Identificar las principales características, ventajas y desventajas de los modelos de AP y su repercusión en la extracción y clasificación de aspectos en diversos dominios.
- Desarrollar un método que combine armónicamente las técnicas de Procesamiento del Lenguaje Natural, el aprendizaje profundo y el Análisis de Sentimientos Basado en Aspectos para garantizar una efectiva clasificación en varios dominios.
- Evaluar la eficacia y eficiencia de los métodos propuestos en la investigación, mediante el uso de medidas como la exactitud, exhaustividad, F1-macro, Cohen-Kappa y el análisis del costo computacional de los modelos.
- Aplicar el método diseñado en la extracción y clasificación de aspectos en la solución de un problema práctico del análisis de sentimientos.

La cantidad de recursos existentes en idioma inglés (p.ej., conjuntos de datos para entrenamiento, diccionarios y herramientas computacionales) para el entrenamiento de modelos de AP permite orientar esta investigación hacia el manejo de opiniones textuales en este idioma.

Las **interrogantes científicas** que se plantea son las siguientes:

- ¿Cómo lograr un método o modelo que combine el aprendizaje continuo y profundo en ABSA y que permita ser empleado en varios dominios del conocimiento, manteniendo una calidad de clasificación alta?
- ¿Cómo definir la forma de representación del conocimiento para que un modelo de aprendizaje profundo en ABSA pueda ser usado durante el proceso de aprendizaje continuo?

Se tienen como objeto de estudio las técnicas computacionales de Inteligencia Artificial para el Procesamiento del Lenguaje Natural y como campo de estudio el procesamiento y análisis de sentimientos en documentos digitales.

La **novedad científica** de la investigación radica en:

- Un nuevo método que combina armónicamente las técnicas de procesamiento del lenguaje natural, aprendizaje profundo, el aprendizaje continuo y el análisis de sentimientos basado en aspectos para garantizar una clasificación efectiva de estos en varios dominios.
- La aplicación de los resultados teóricos de la investigación a través de un sistema creado por el autor y varios colaboradores para gestionar la información de campañas de bien público.

El **aporte práctico** del trabajo está dado por:

Un modelo computacional para la clasificación, que puede ser empleado en sistemas de toma de decisiones, y que permite ser entrenado con nuevos conjuntos de opiniones con niveles altos o similares de calidad a los obtenidos anteriormente.

Se define como **hipótesis de investigación**:

Un modelo computacional que combina una arquitectura de aprendizaje profundo centrada en mecanismos de atención, y un modelo de aprendizaje continuo que siga una estrategia de regularización en la subtarea de análisis de sentimientos basado en aspectos, logra una clasificación **eficaz y eficiente** de aspectos en documentos en idioma inglés provenientes de diversos dominios.

La **tesis** está estructurada en tres capítulos. En el primer capítulo se analizan los modelos de aprendizaje profundo con mejores resultados de calidad en la subtarea ABSA; se mencionan las posibilidades y limitantes de cada uno de ellos. Se describen y evalúan, además, las diversas formas de representación del conocimiento que usan estos modelos y se estudian los conceptos del AC, su estrategia de regularización, con sus principales propuestas.

En el segundo capítulo se presentan dos nuevas propuestas: la primera, para la extracción de aspectos, plantea la integración de un modelo clásico de AP y otro de AC; la segunda, tiene como objetivo la clasificación de aspectos y combina un modelo de AP de tipo *Trasformer* con uno de AC. De igual manera, se analizan los resultados teóricos alcanzados y su evaluación.

En el tercer capítulo se aplican los resultados teóricos de la investigación en la plataforma WisePocket para la gestión de campañas de bien público. Se presentan ambos, la plataforma y la manera en que se usan los aportes de esta investigación en la toma de decisiones sobre los resultados de las campañas. Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas, producción científica del autor y los anexos.

# CAPÍTULO 1

## Acerca del análisis de sentimientos basado en aspectos

El procesamiento de las opiniones es muy importante para la toma de decisiones en las empresas, instituciones, organizaciones, o por los individuos [4]. La información que se obtiene por ejemplo, la manera de mejorar un producto, los problemas de salud que afectan a la población, entre otras, permite determinar el éxito comercial de una empresa o salvar vidas.

El AP es un concepto que ha ganado auge en los últimos años por su efectividad en la creación de modelos computacionales de alta eficacia en varios problemas de la vida real (clasificación de imágenes, audio o análisis de sentimientos, entre otros) [42]. Una de las principales ventajas de los modelos de AP es que no necesitan de la ingeniería de características para reconocer patrones en flujos de datos [17]. La información textual, que conforman las opiniones, necesita ser transformada en otra forma de representación que permita el entrenamiento de las redes neuronales de los modelos de AP [43, 44, 45].

El Análisis de Sentimientos Basado en Aspectos (*Aspect Based Sentiment Analysis*; ABSA) es la subtarea del Procesamiento del Lenguaje Natural (PLN) [46] que más información ofrece para la toma de decisiones asociadas a flujos de opiniones, por el grado de especificidad que proporciona [4].

Para lograr que los modelos computacionales de AP puedan predecir la información de flujos de opiniones de diferentes dominios (e.d., opiniones sobre hoteles, restaurantes, elecciones presidenciales), es necesario crear modelos de AC [36]. En el PLN estos modelos son un campo de investigación incipiente [25].

En este capítulo se describen los conceptos asociados al ABSA, se abordan las formas de representación del texto digital para el descubrimiento de conocimiento en modelos de AP, y se particulariza en los principales modelos de AP en la subtarea de ABSA y del AC.

## **1.1 Características generales del análisis de sentimientos basado en aspectos**

---

### **1.1. Características generales del análisis de sentimientos basado en aspectos**

Una palabra identificada como “aspecto” a clasificar en la información textual de una opinión, está relacionada con una o varias entidades presentes en una estructura de información (p.ej., grupos de palabras, oración o documento). En la oración “*La comida de ese restaurante es deliciosa y barata.*”, son entidades las palabras “*comida*” y “*restaurante*”. La identificación es una tarea muy importante del PLN y compleja, ya que se encarga de determinar la categoría a la que pertenece la entidad (lugar, organización, evento, etc.) y también de relacionar las posibles referencias a una misma entidad en un texto, para lograrla se pueden usar modelos computacionales o bibliotecas [46]. Esta investigación no profundiza en su estudio, y durante el pre-procesamiento del texto digital se utilizan herramientas de terceros para su identificación.

Las opiniones sobre los aspectos o características de una entidad pueden revelar mayor información. Por ejemplo, en una opinión sobre un producto o una enfermedad, la persona que la ofrece refiere datos positivos o negativos. Los aspectos son palabras (p.ej., sustantivos, adjetivos, adverbios, etc.) y pueden estar asociados a otras palabras (negación, adverbios, conjunciones) que pueden variar el sentimiento expresado en ellos [4].

El ABSA permite mayor detalle de los sentimientos expresados por el autor o autores de un texto analizado [10, 47, 48]. Este está formado por dos subtareas principales:

- Extracción de aspectos: Se encarga de extraer aspectos y entidades de los documentos teniendo en cuenta que estos pueden ser explícitos o implícitos. Por ejemplo, en la oración “*La comida de ese restaurante es deliciosa y barata.*”, tiene como aspecto “*la comida*” de la entidad “*restaurante*”. En esta oración el adjetivo “*barata*” expresa la existencia de un aspecto implícito que es el “*precio*”.

Esta investigación se centra en la extracción y clasificación de aspectos explícitos en oraciones (e.d., una palabra simple o una frase) y que se diferencian de las entidades nombradas.

- Clasificación de los sentimientos del aspecto: Determina si la opinión emitida sobre el aspecto es positiva, negativa o neutral. En el ejemplo anterior la opinión sobre la comida

## 1.1 Características generales del análisis de sentimientos basado en aspectos

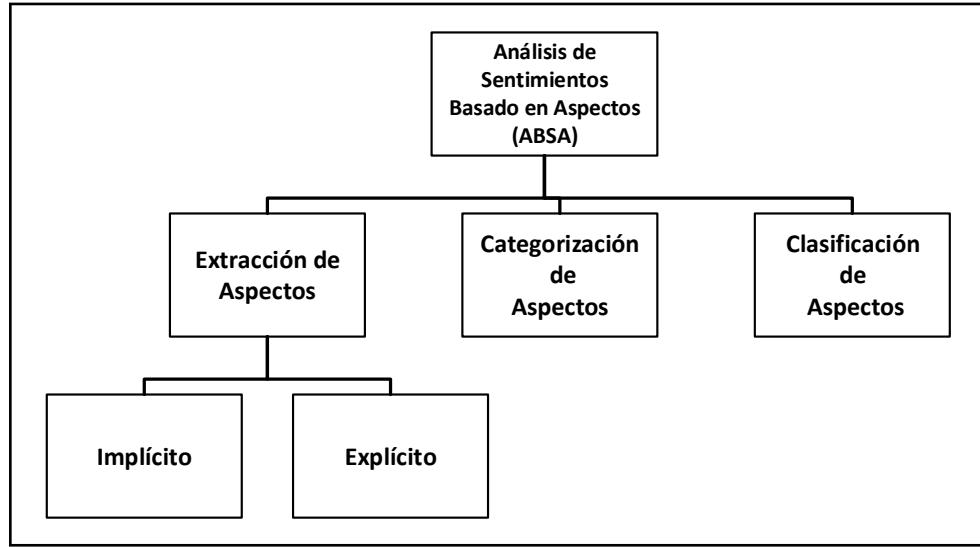


Figura 1.1: Taxonomía de subtareas para ABSA, tomado de [1] .

del restaurante es positiva<sup>1</sup>.

En la propuesta presentada en [49] se establecen tres importantes subtareas para el ABSA:

- Extracción del objeto de la opinión (*Opinion Target Expression*; OTE) o extracción de aspectos: tiene como objetivo la extracción de los términos o palabras clasificadas como aspectos (p.ej., entidad o atributo). Está asociado a la extracción de aspectos.
- Detección de la categoría del aspecto (*Aspect Category*; AC) o categorización de aspectos: se relaciona con la identificación y agrupamiento de los aspectos en conceptos más generales como *comida*, *confort*, *limpieza*, etc.
- Polaridad del sentimiento (*Sentiment Polarity*; SP) o Clasificación de Aspectos: establece un sentimiento a los aspectos extraídos (p.ej., positivo, negativo o neutral).

Por ejemplo, en la oración “*La comida de ese restaurante es deliciosa y barata.*” los aspectos “*deliciosa*” y “*barata*” son obtenidos por un modelo computacional que resuelva la subtarea OTE. En el caso de la subtarea AC estos aspectos pudieran ser clasificados como comida u otro concepto que los agrupe, mientras que la subtarea SP daría a estos aspectos una polaridad positiva [1, 5, 10].

<sup>1</sup>En algunos trabajos como en [33] pueden ser de una especificidad mayor como: fuertemente positivo, positivo, débilmente positivo, neutral, fuertemente negativo, negativo y débilmente negativo.

Algunos autores [13, 14, 50] han tratado de proponer modelos para ABSA con el uso de reglas lingüísticas o diccionarios de palabras, pero la construcción de estos recursos es muy costosa, por lo que una alternativa para la extracción y clasificación de aspectos ha sido el empleo de diferentes modelos computacionales como el AP, sobre conjuntos de entrenamiento en dominios específicos.

Los conjuntos de datos son un recurso importante para los modelos computacionales, porque son necesarios en su entrenamiento y evaluación (p.ej., audios, opiniones textuales, entre otros) [51]. En el caso del análisis de sentimientos, la construcción de estos conjuntos incrementa su costo cuando se desea extender a otros idiomas como el español [4].

La existencia de una mayor cantidad de conjuntos de datos en idioma inglés y recursos asociados (p.ej., etiquetadores morfológicos, diccionarios, etc.) permitió que esta investigación se orientara a la creación de modelos computacionales para este idioma.

## 1.2. Clasificación textual

Existen varias estrategias para la extracción y clasificación de aspectos en ABSA:

- Empleando la frecuencia con que aparecen los términos y el uso de diccionarios o lexicones [13, 43].
- Analizando las relaciones sintácticas [52].
- Usando técnicas de aprendizaje supervisado como Campos Aleatorios Condicionales (*Conditional Random Field; CRF*) [53, 54] y Máquinas de Vectores de Soporte (*Support Vector Machine; SVM*) [55].
- Empleando métodos no supervisados, basados en la detección de tópicos presentes en un documento según la Asignación Latente de Dirichlet (*Latent Dirichlet Allocation; LDA*) [56] y otras propuestas derivadas de esta [57, 58].

La efectividad de estas estrategias depende de la calidad de las características seleccionadas en los conjuntos de datos de entrenamiento o de los recursos usados para la creación de lexicones, diccionarios u ontologías [43, 59]. Esto hace muy costoso el proceso de creación de conjuntos de datos que sirvan para el entrenamiento de modelos de aprendizaje automatizado en ABSA.

El aprendizaje automático de modelos computacionales de forma supervisada tiene como objetivo que en el proceso final el resultado de estos modelos se ajuste bien a los datos del conjunto de entrenamiento; si ha sido seleccionada la estrategia correcta [51, 60, 61]. Para vencer este reto se han presentado en la literatura varias propuestas. Una de las primeras fue la presentada por [16].

En la misma se usa una optimización Bayesiana basada en el Descenso del Gradiente, calculando la probabilidad condicional de cada palabra  $w$  a pertenecer a una de las clases  $c_i$ . En este trabajo no se tiene en cuenta la relación que pudiera tener  $w$  con el contexto que la rodea en una oración o documento (p.ej., la aparición de términos de negación antes o después de la palabra, diferentes significados en cada dominio). Esto influye en los resultados de desempeño del modelo entrenado, porque no le permite reconocer el valor semántico que tiene la polaridad de  $w$  en el texto (p.ej., en “la camisa no es bonita” no tener en cuenta la influencia de la negación en la vecindad de la palabra “bonita” provoca que el modelo prediga que es un opinión positiva sobre la camisa.).

La propuesta de [62] presenta un modelo no supervisado de aprendizaje en múltiples dominios de forma continua para la clasificación del objetivo de las opiniones entre entidades nombradas y aspectos. Este modelo tiene como principal estructura computacional un grafo que almacena las relaciones entre aspectos, entidades nombradas y su contexto. Para encontrar estas relaciones y nuevos patrones se usa un conjunto de reglas lingüísticas.

La propuesta presenta como resultado un F1-macro de 0.79 y permite considerar el uso de reglas lingüísticas como parte del proceso de aprendizaje de un modelo. Pero el proceso de entrenamiento del modelo solo se realizó en conjuntos de datos o dominios de efectos electrodomésticos, lo que no permite determinar su desempeño durante el aprendizaje de dominios diferentes (p.ej., opiniones de hoteles). Durante la evaluación de la propuesta no se determina si existe olvido catastrófico durante el AC, lo cual no permite analizar si se mantiene el conocimiento aprendido entre dominios.

En [63] se presenta un modelo basado en Campos Condicionales Aleatorios (*Conditional Random Fields*; CRF) para mejorar la extracción de aspectos de forma supervisada. Una de las principales desventajas de esta propuesta es el uso del CRF, debido a que este modelo necesita de un conjunto de características (p.ej.: etiquetas morfológicas de los posibles aspectos, relación con otras palabras del conjunto de entrenamiento) que hacen costoso encontrar conjuntos

de datos o especialistas capaces de construirlos.

El proceso de evaluación del modelo se realiza con conjuntos de datos de dispositivos electrónicos, que no permite determinar su capacidad de aprender en dominios diferentes; la manera de estimar la efectividad del modelo es mediante el promedio de los resultados de la efectividad en cada dominio (e.d., precisión, exhaustividad y F1-macro). Esto puede ocultar un alto olvido catastrófico porque no se analiza cómo es el aprendizaje al pasar de un dominio a otro.

Uno de los conceptos con mucho éxito, al aplicarlo a varios dominios del conocimiento humano (p.ej., procesamiento de imágenes y del lenguaje natural), es el AP presentado por [17, 19, 42]. Dos de sus más destacados aportes son: reduce la necesidad de construcción de los datos mediante un pre-procesamiento inicial y elimina la necesidad de realizar ingeniería de características en los conjuntos de entrenamiento [17, 19]. Durante este proceso también pueden aparecer características no detectadas por los humanos [17].

El AP agrupa técnicas y modelos de aprendizaje tanto supervisados como no supervisados; todos tienen como estructura principal varias capas de Redes de Neuronas Artificiales (*Neural Networks*; NN) [19], capaces de aprender una representación jerárquica en arquitecturas profundas [17].

La capa final de la red representa la predicción del modelo [17, 19]. La función de pérdida determina la exactitud de esta predicción calculando el error entre los valores obtenidos en la última capa y los valores de comprobación. Un algoritmo de optimización como el Descenso Estocástico del Gradiente (*Stochastic Gradient Descent*; SGD) [64] es empleado para ajustar los pesos de las neuronas, durante su propagación hacia atrás, calculando el gradiente de la función de pérdida [65].

La exploración y explotación del espacio de posibles soluciones se logra porque el proceso de aprendizaje en la arquitectura de red neuronal repite varias veces el ciclo o *epoch* de entrenamiento sobre los datos, hasta que el error alcanza una cota deseada [17]. En el entrenamiento de las redes neuronales de los modelos de AP, un parámetro importante (o hiper-parámetro) es el tamaño del lote o *batch*. Este hace referencia a los subconjuntos de datos del entrenamiento, que son tomados en cada iteración para ser evaluados. Esta es una optimización que permite rápidamente converger hacia una solución, al tener en cuenta a la vez, durante el entrenamiento, varias instancias o ejemplos y su valor objetivo [17, 19], debido a que en un menor tiempo es posible evaluar una mayor cantidad de instancias (e.d., ejemplos) del problema a enfrentar.

Algunos modelos de AP han mejorado su desempeño a partir de:

- El uso de Unidades de Rectificación Lineal (*Rectified Linear Units*; ReLUs) como funciones de activación [66].
- La introducción de métodos de marginación (*dropout*) [67].
- La inicialización aleatoria de los pesos de las redes [68].
- La solución del problema de la desaparición del gradiente, así como su explosión (p.ej., valores en los pesos de las neuronas menores que cero o que tienden al infinito) con el uso de redes de memoria de corto plazo (*Long Short-Term Memory*; LSTM) [19].

El AP agrupa varios modelos que definen la arquitectura de las redes neuronales que los conforman [17, 19]. En la Tabla B.1, en el Anexo B, se muestra una descripción de modelos de AP usados en trabajos sobre ABSA.

En la figura 1.2 se muestra una taxonomía, creada a partir de esta investigación, que clasifica los principales métodos de aprendizaje profundo estudiados. La clasificación correspondiente a los “Modelos de Aprendizaje Profundo” hace referencia a aquellos artículos que utilizan Redes Neuronales Convolucionales (*Convolutional Neural Network*; CNN), Memoria de Corto Plazo (*Long Short Term Memory*; LSTM) o Unidades Recurrentes con Compuertas (*Gated Recurrent Unit*; GRU) para la subtarea ABSA [69, 70, 71].

La clasificación nombrada “Mecanismo de Atención + Modelos de Aprendizaje Profundo” agrupa aquellos trabajos que combinan el Mecanismo de Atención con modelos como CNN, LSTM, redes de memoria de corto plazo bidireccionales (*Bidirectional Long Short-Term Memory*; BLSTM) o Unidades Recurrentes con Compuertas Bidireccionales (*Gated Recurrent Unit*; BGRU) [72, 73]. Esta combinación se ha convertido en una práctica muy extendida entre los investigadores en ABSA [5, 8].

La clasificación “Máquinas de Aprendizaje + Modelos de Aprendizaje Profundo” se refiere al uso de modelos de AP como CNN, LSTM o BLSTM y luego una máquina de aprendizaje de tipo CRF [74, 75]. Los trabajos analizados emplean el aprendizaje profundo a través de enfoques supervisados [17].

### 1.3 Formas de representación textual

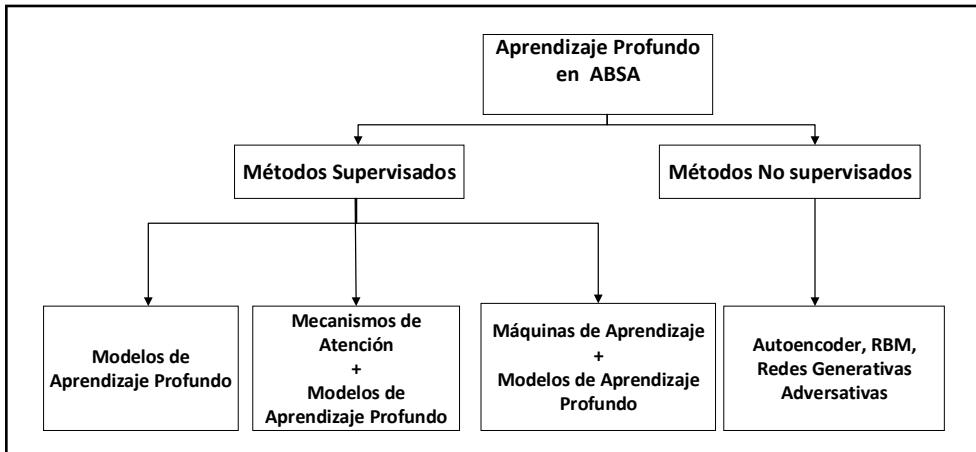


Figura 1.2: Taxonomía de métodos de Aprendizaje Profundo para ABSA.

## 1.3. Formas de representación textual

Los textos generalmente se conforman por párrafos, oraciones y palabras. Para lograr un desempeño efectivo de los modelos de AP, y consecuentemente realizar una extracción y clasificación eficaz de aspectos en ABSA, es necesario realizar una organización correcta de la información no estructurada [5].

El AP se basa, esencialmente, en el trabajo con redes neuronales y varios métodos han surgido para transformar cadenas de caracteres en datos numéricos que son, por lo general, la entrada de estas redes [76].

Una estrategia consiste en tomar el vocabulario de palabras en el conjunto de documentos que se desean analizar y realizar una representación de cada una usando un vector de una aparición (*one-shot*). Este método tiene varias desventajas: en primer lugar, no es escalable si el tamaño del vocabulario es muy grande; en segundo lugar, muchas palabras son equidistantes y no guardan ninguna relación semántica con su contexto (porque cada una de ellas solo ocupa una dimensión) [77].

Otra estrategia es la Frecuencia Inversa por Documento (*Inverse Document Frequency*; TF-IDF) [15], que asigna una relevancia mayor a aquellos términos que aparecen en pocos documentos, y una importancia menor a los que están presentes en la mayoría, proporcionando una medida de importancia en relación con todo el *corpus* o conjunto de entrenamiento.

## 1.3 Formas de representación textual

---

Esta forma de representación, aunque usada por varios trabajos en ABSA [78, 79], no mantiene la relación semántica entre las palabras y su contexto, muy importante en ABSA donde el grado de polaridad o subjetividad puede ser determinado a partir de las palabras que se encuentran en la vecindad analizada [80]. Varios modelos han tratado de superar esta limitación y son los siguientes:

### 1.3.1. Vector de palabras embebidas (*Word Embeddings*)

La necesidad de encontrar formas de representación, que permitan conservar la relación semántica de las palabras en el texto, permitió el surgimiento de un modelo conocido como vector de palabras embebidas(*Word Embeddings*) [31, 81] del cual existen varios ejemplos:

**C-W:** Aprende un modelo de lenguaje sobre un gran conjunto de datos para obtener las relaciones semánticas y sintácticas de las palabras. Optimiza la pérdida de la entropía cruzada para maximizar la probabilidad de una palabra dada con respecto a las palabras previas de una secuencia [81].

**Word2Vec:** Usa red neuronal simple y permite durante el entrenamiento incrementar la velocidad de aprendizaje y la calidad de las palabras embebidas generadas [26]. Para este se definen dos estrategias: la **Bolsa de Palabras Continuas (Continuous Bag-of-Words; CBOW)**: [26] y **Skip-Gram**: [80].

**Vectores Globales (Global Vectors; Glove):** Este modelo propuesto en [27] tiene algunas diferencias con el que aparece en [26]. Glove realiza una estadística de co-ocurrencia de las palabras que no es explícita en Word2Vec, que realiza el entrenamiento iterativo de una red neuronal simple [27].

Una diferencia importante entre los modelos propuestos por Word2Vec es que estos deben ser reentrenados si se desea cambiar la dimensionalidad del vector embebido. Sin embargo, Glove puede reusar la matriz de co-ocurrencia ya calculada.

**FastText:** El modelo propuesto en [28] es una extensión del modelo Skip-Gram de Word2Vec. Cada palabra es representada por una bolsa de los  $n$ -gramas de los caracteres y la palabra misma. En lugar de una sola palabra objetivo  $w_t$ , en este modelo es usada la bolsa de  $n$ -gramas para entrenar el modelo Skip-Gram.

**Modelos de Lenguaje Embebidos (Embeddings from Language Models; ELMo):** El mo-

delo propuesto en [30] se centra en las palabras embebidas del contexto, a diferencia de Word2Vec y FastText. Para codificarlas se entrena un modelo de lenguaje bidireccional profundo en un conjunto de datos grande. Los estados internos de este modelo son usados para calcular el Word Embeddings. El vector de representación de una palabra es una función de la secuencia de entrada (oración). Este modelo usa también la información de los caracteres de la palabra como en el FastText; puede usar la información presente en  $n$ -gramas y estimar la representación de las palabras fuera del vocabulario.

### 1.3.2. Representación de Codificador Bidireccional de Transformadores *(Bidirectional Encoder Representation with Transformer; BERT)*

BERT es un método de representación del lenguaje natural [31] y su arquitectura de la red neuronal es un codificador bidireccional multi-capa<sup>2</sup>. Su modelo computacional se basa en una capa de auto-atención [82] en vez de usar capas de redes recurrentes. Esto es debido a que la auto-atención ofrece mejores posibilidades de ser paralelizada y de reducir el costo computacional con respecto a las LSTM [77, 82].

Una de sus diferencias con respecto a los modelos CBOW, Skip-Gram, Glove y FastText radica en su forma de entrenar la red neural, porque adiciona a la palabra la posición donde esta aparece en la capa de entrada del modelo. Otra consiste en que realiza simultáneamente el entrenamiento para la predicción de las palabras de izquierda a derecha y de derecha a izquierda, donde la palabra que se predice se posiciona al final o al inicio de la secuencia de texto.

BERT puede ser ajustado fácilmente, añadiendo una capa adicional a la salida del mismo que no necesita ser entrenada para una gran cantidad de parámetros, debido al pre-entrenamiento hecho con los modelos de BERT [31] lo cual permite no necesitar grandes cantidades de datos para un nuevo modelo que use como entrada la salida de BERT. Esto es muy importante en la subtarea ABSA, porque los conjuntos de datos de entrenamiento, en su mayoría, no cuentan con grandes cantidades de ejemplos o instancias.

Los argumentos para seleccionar a BERT como parte de la arquitectura del modelo propuesto en esta investigación son los siguientes:

- El uso de un modelo BERT pre-entrenado permite obtener como entrada al modelo de

---

<sup>2</sup>Implementado en la biblioteca propuesta en tensor2tensor <https://github.com/tensorflow/tensor2tensor>

AP un vector, y de esta manera asociar los *tokens* presentes en una oración con los elementos de clasificación.

- Varias propuestas para la subtarea ABSA han mostrado mejoras en su desempeño, al hacer uso de BERT como en [32, 83, 84].
- La existencia de modelos pre-entrenados para el idioma inglés.

### 1.4. Extracción y clasificación de aspectos

Herramientas importantes en el procesamiento de la información y el PLN son los modelos supervisados, porque permiten realizar la identificación de palabras que representan aspectos en el texto digital, y su posterior clasificación según su polaridad (e.d., positivo, negativo u otros valores). En este epígrafe se analizan varios modelos supervisados de Aprendizaje Profundo utilizados en ABSA para la extracción y clasificación de aspectos.

#### 1.4.1. Redes Neuronales Convolucionales (*Convolutional Neural Network; CNN*)

La capa de convolución o *kernel* es la parte principal de una CNN. Consiste en un conjunto de parámetros a aprender llamados filtros; poseen la misma forma que la entrada, pero de menor dimensión [17, 19, 42]. Esta arquitectura de red permite, a partir de la representación de las palabras, aplicar en cada capa de la red una operación de convolución (selección de características importantes).

La complejidad computacional de cada capa de un CNN es  $O(k \cdot n \cdot d^2)$ , donde  $k$  es el tamaño del kernel,  $n$  es el tamaño de las secuencias de vectores de entrada y  $d$  es la dimensión de cada vector [82]. Una desventaja de esta estrategia es que el tamaño de la entrada a la red no puede ser variable, en el caso del PLN las oraciones o conjuntos de palabras cortas deben completar con ceros los espacios restantes, lo que incurre en un gasto adicional de memoria.

La figura 1.3 muestra la arquitectura de una CNN, cuya salida (e.d., compuesta por tres neuronas  $x, y, z$ ) puede estar asociada en ABSA a los valores de clasificación positivo, neutral y negativo.

En [34, 35, 85] se emplea una variante de CNN para ABSA, mediante el uso de una secuencia

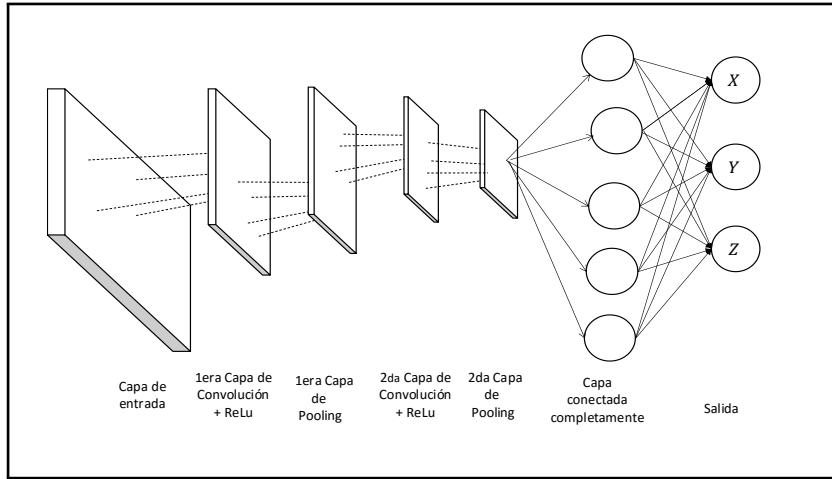


Figura 1.3: Arquitectura de una CNN de siete capas.

de redes convolucionales donde la salida de una red es la entrada de la otra. La selección de estos tipos de algoritmos por parte de los investigadores se justifica por la variedad de problemas de PLN que se pueden resolver aplicando las CNN y los buenos resultados de calidad (e.d., a partir de los valores de medidas como la exactitud y el F1-macro) que se han obtenido [1].

En [86] se relaciona la salida de un CNN con un CRF, pero no mejora los resultados de SVM contra los que se prueba este método; los resultados mostrados en los artículos analizados indican que se debe revisar si realmente es beneficioso relacionar métodos de aprendizaje profundo con otros modelos, tales como CRF o aumentar el conjunto de datos con el que es entrenado el modelo CNN.

### 1.4.2. Redes Neuronales Recurrentes (*Recurrent Neural Network; RNN*)

Están orientadas a resolver problemas de secuencias o de series de tiempo (p.ej., audio, texto) [87], que pueden tener tamaño variable. La entrada de una RNN consiste en el dato actual y el anterior (p.ej., la representación del *token* o palabra en una oración y la que la antecede). Significa que la salida en el instante  $t - 1$  afecta la entrada del instante  $t$ . Cada neurona está equipada con un ciclo de retroalimentación que retorna la salida actual como entrada del próximo paso, como se muestra en la figura 1.4. El conjunto de neuronas que procesa la información del dato actual y del anterior es conocida como capa hacia adelante [8, 42].

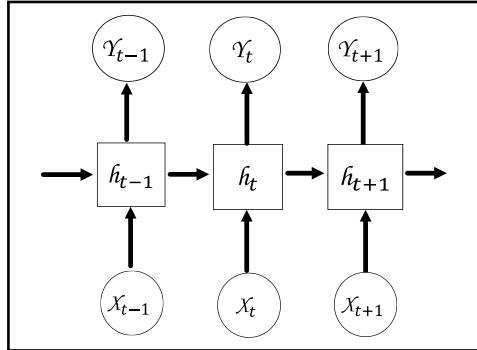


Figura 1.4: Arquitectura de una RNN de tres capas.

Dos desventajas de las RNN son la desaparición del gradiente, asociado a que este sea cercano o igual a cero, o su incremento considerable [8, 17, 19].

Existen diferentes tipos de RNN, como son: la **Memoria de Corto Plazo** (*Long Short Term Memory*; LSTM) [19], que resuelve los problemas de la desaparición y explosión del gradiente de las RNN e introduce el concepto de compuertas para las neuronas; las **Unidades Recurrentes con Compuertas** (*Gated Recurrent Unit*; GRU) [17] se diferencian principalmente de las redes LSTM en que solamente definen dos compuertas y las hacen menos costosas computacionalmente en cantidad de parámetros (memoria) y en el tiempo de entrenamiento de la red neuronal.

Los tres modelos anteriores se enfocan en obtener el próximo estado a partir del anterior. Una propuesta que ha obtenido buenos resultados en ABSA con el modelo RNN, consiste en incorporar una capa hacia adelante y otra hacia atrás para aprender la información de los *tokens* próximos y anteriores (e.d., una capa que tiene en cuenta el incremento de la información en la serie de tiempo y otra del decrecimiento), conocidas como **Redes Recurrentes Bidireccionales** [88].

En algunos trabajos analizados se realiza una hibridación mediante la salida de un método de AP con el entrenamiento de una máquina de aprendizaje como (*Conditional Random Fields*; CRF). En [75] inicialmente se entrena una RNN, y la salida resultante se utiliza como entrada de un CRF. Esta propuesta no supera los resultados del método ganador en la competencia SemEval 2016 para la subtarea ABSA.

En [86] se propone el uso de LSTM y en la salida de la red neuronal un CRF para el aprendizaje. Con el empleando en esta variante se obtuvieron resultados de más de un 0.83 de F1-micro.

## **1.4 Extracción y clasificación de aspectos**

---

Aunque el uso de LSTM tiene el objetivo de reducir las características a entrenar por el modelo, aún se imponen las desventajas de usar un modelo CRF y la necesidad de tener en cuenta la ingeniería de características al construir o seleccionar los conjuntos de datos a entrenar.

En [89] se compara el modelo propuesto con los mejores métodos de la competición SemEval 2014 para la subtarea ABSA. Aunque no obtuvo los mejores resultados fue tomando en cuenta en esta investigación porque demostró la posibilidad de emplear AP para ABSA.

### **1.4.3. Mecanismos de Atención**

En ABSA, las propuestas que usan Mecanismos de Atención (MA), conocida en la literatura como (*Attention Mechanism; AM*), permiten al modelo aprender centrando la atención en la interacción entre el aspecto y la palabra o *token* de la oración que más relación tiene con este [90] (p.ej., las palabras más cercanas al aspecto). La idea general consiste en calcular un peso de atención a partir del vector embebido y usando la función *Softmax* [19], para lograr una representación de mayor nivel [90].

El MA mejora el problema de las RNN de codificar información irrelevante, en especial cuando la entrada es muy rica en información [44]. Esto se debe a que al usar el MA tienen mayor importancia las palabras o tokens más cercanos en vencidad (e.d., grupo de palabras cercanas por la izquierda y derecha de una palabra), lo cual permite preservar la relación semántica existente en el contexto donde aparece una palabra (p.ej. un aspecto en una oración).

Entre sus ventajas se tiene que han sido aplicadas con éxito a varias tareas del PLN, así como su habilidad para capturar partes importantes de un texto. En MA la principal ecuación que captura la relación de una palabra o conjunto de palabras es:

$$\alpha_i = softmax(h_i) = \frac{exp(score(h_i, a_j))}{\sum_j exp(score(h_j, a_j))} \quad (1.1)$$

Donde  $a_j$  representa el componente objetivo (p.ej.; un aspecto) a entrenar en el modelo,  $h_i$  un elemento del contexto o conjunto de vectores asociados y  $score(h_i, a_j)$ ) calcula una puntuación entre  $h$  y  $a_j$ . El valor de  $\alpha_i$  indica la relación semántica que existe entre el contexto y el aspecto  $a_j$  y que es usado durante el aprendizaje de modelos de AP.

### 1.4.4. Redes de Memoria (ReM)

Este modelo está dividido en dos partes: representación del aspecto y representación del contexto. La primera es el vector embebido de la palabra que constituye el aspecto. Los vectores del contexto (*token* de la oración o el texto)  $m_1, m_2, \dots, m_n$  son apilados y convertidos en una memoria externa  $m$ . El estado interno  $\mu$  se establece como la representación del aspecto. La coincidencia entre  $\mu$  y cada memoria  $m_i$  es calculada por el producto interno y una capa de *Softmax* [19], que constituye la representación del contexto. El resultado es un vector de probabilidad  $p$  [8]. El uso de la capa *Softmax* incluye en este modelo de AP el uso de MA.

En [21, 91] se usa una Red de Memoria Profunda (*Deep Memory Network*; DMN) que es entrenada a partir de un conjunto de aspectos predefinidos. En varios trabajos [21, 22, 91] se reporta el empleo del MA que promedia los pesos relevantes en otros puntos de una red neuronal. Este mecanismo permite incluir características lingüísticas o sintácticas en el proceso de aprendizaje de la red neuronal que implementa el modelo de AP. Varias propuestas [35, 70, 88] agregan reglas lingüísticas al uso de modelos de AP.

En [91] se hace una hibridación de LSTM y MA, que es entrenada a partir de un conjunto de aspectos predefinidos. El vector asociado al aspecto y los vectores de las palabras del contexto son empleados para aprender características, y se obtiene la distancia entre aspectos usando un agrupamiento K-means. Sus resultados superan a otros métodos de ABSA, pero se establecen como medidas de eficacia la *pureza* y la *entropía*, que no son usadas por la mayoría de trabajos sobre AP y ABSA [5, 8]. Esto dificulta la comparación de la efectividad de este método con otras propuestas recientemente publicadas.

Los resultados relacionados con la eficacia de la hibridación no superan los modelos de aprendizaje automatizado como CRF o Máquinas de Vectores de Soporte (*Support Vector Machine*; SVM), sin embargo, el uso de AP se relaciona con la posibilidad de no necesitar la ingeniería de características para el entrenamiento de los modelos de AP y lograr que los modelos CRF y SVM tengan como entrada conjuntos de características reducidos; la mejora de su eficacia, permite su entrenamiento a partir de datos en bruto.

Algunas propuestas no tienen como objetivo extraer y clasificar todos los aspectos, sino que extraen los asociados a categorías prefijadas [92, 93, 94]. Por ejemplo, en [94] un conjunto de datos sobre restaurantes se evalúa para determinar cuán buena es la clasificación de aspectos o palabras asociadas a *comida*, *personal* y *ambiente*.

## **1.4 Extracción y clasificación de aspectos**

---

Adicionalmente, han aparecido en la literatura propuestas de paralelización con el fin de reducir el costo temporal [59, 69, 94].

También aparecen más trabajos con modelos supervisados que no supervisados o híbridos [5, 8, 44], esto se debe a la existencia de conjuntos de datos que permiten la creación y entrenamiento de modelos en forma supervisada. Aproximadamente, el 42 % corresponden a variantes que usan LSTM como modelo principal de AP [5].

Los modelos supervisados sufren la desventaja de necesitar para su entrenamiento muchos datos o ejemplos etiquetados [95, 96]. Una posible solución es el uso de estrategias como el aprendizaje de pocas instancias (*Few-shot Learning*) [97, 98]. En [99] se propone resolver la subtarea de detección de la categoría del aspecto, con la estrategia Few-shot Learning. Aunque los valores de exactitud (p.ej., 0.73 para conjuntos de datos de opiniones de restaurantes) en [99] no son mejores que para otras propuestas, muestra la validez del Few-shot Learning. Este tipo de enfoque pudiera ser empleado en idiomas como el español, en el que existen pocas propuestas y recursos para ABSA [5].

### **1.4.5. Grafos Convolucionales Neuronales (GCN)**

Para superar algunas deficiencias de las RNN y CNN para ABSA, se han propuesto los Grafos Convolucionales Neuronales (GCN), como una generalización de las redes neuronales recursivas [100, 101, 102].

En un GCN, para todo nodo del grafo (que representa los *tokens* de una oración o documento), se codifica la información relevante sobre su vecindad como un nuevo vector de representación. En ABSA, los *tokens* y los aspectos son tratados como nodos, y una arista es la relación de dependencia del sentimiento entre dos nodos [23, 103]. En [1] se resume que los GCN con frecuencia muestran el mejor desempeño con dos capas, y GCN más profundos no muestran mejores resultados debido al problema del sobreentrenamiento.

En [104] se usan los GCN con el modelo pre-entrenado BERT. La combinación de estas estrategias (e.d., la información semántica que proporciona BERT y los GCN) alcanza valores de exactitud de 0.90 y de F1-macro de 0.74, para un conjunto de datos de entrenamiento de opiniones de restaurantes, que son valores muy altos en el estado del arte [1].

Los modelos de AP, en su mayoría, han sido entrenados de forma supervisada en un solo domi-

## **1.5 Propuestas de modelos de aprendizaje continuo**

---

nio del conocimiento (e.d., opiniones de restaurantes, opiniones en Twitter, etc.). Sin embargo, después de crear un clasificador en uno de esos dominios y usarlo en el entrenamiento de uno nuevo, los pesos (el conocimiento de la red) se pierden con la nueva información aprendida. Se necesitan modelos que permitan, en un AC, preservar los pesos anteriores.

### **1.5. Propuestas de modelos de aprendizaje continuo**

El desarrollo de sistemas que emplean el AC con modelos de AP ha tenido un interés creciente en los últimos años [25]. En estos modelos es importante mantener la efectividad en todos los dominios o tareas y disminuir o evitar el olvido catastrófico [16, 38].

En este epígrafe se anuncian las bases conceptuales que definen el AC para modelos de AP y su relación con la subtarea ABSA. Luego se describen varias estrategias de AC usadas en ABSA y se analizan sus principales ventajas y desventajas.

La creación de modelos de AC trata de imitar la capacidad de los seres humanos durante el proceso de aprendizaje de nuevas tareas. Estos comprenden y aprenden nuevos conceptos auxiliados por sus memorias pasadas y aplicando el conocimiento y las experiencias anteriores (p.ej., mediante la analogía o asociación) [36, 38, 105, 106].

Los modelos computacionales que usan redes neuronales suelen ser entrenados para ser efectivos en una tarea o dominio específico; comienzan a ser menos precisos en el tiempo, como consecuencia, por ejemplo, del cambio de la distribución de los datos [38, 107, 108, 109].

El AC también es conocido como Aprendizaje Constante (*Lifelong Learning*) [16], Aprendizaje Secuencial (*Sequential Learning*) [37] o Aprendizaje Incremental (*Incremental Learning*) [110].

Se define el aprendizaje continuo o constante en [16] donde aparece como:

En un proceso de aprendizaje continuo, en cualquier momento del tiempo, el modelo de aprendizaje ha ejecutado una secuencia de  $N$  tareas de aprendizaje,  $\tau_1, \tau_2, \dots, \tau_N$ . Estas tareas, llamadas *tareas previas*, tienen sus correspondientes conjuntos de datos  $D_1, D_2, \dots, D_N$ . Pueden ser de diferentes *tipos* y diferentes *dominios*. Cuando el modelo se enfrenta a la tarea  $\tau_{N+1}$  (la cual es llamada la *nueva tarea* o *tarea actual*) con su conjunto de datos  $D_{N+1}$ , el modelo de aprendizaje

## 1.5 Propuestas de modelos de aprendizaje continuo

---

puede usar el conocimiento aprendido en las tareas anteriores en la *base de conocimientos* (BC) para ayudar en el aprendizaje de la tarea  $\tau_{N+1}$ . La BC es actualizada al terminar el aprendizaje de  $\tau_{N+1}$  sin eliminar el anterior.

En este trabajo se asume esta definición, aunque se aclara la necesidad de definir las medidas de desempeño  $P$  sobre una función objetivo  $h^*$  que se asocia a la función de pérdida de las redes neuronales profundas.

Para una definición formal de aprendizaje continuo se define:

En un modelo de aprendizaje continuo, dada una secuencia de tareas o dominios  $D_1, D_2, \dots, D_T$  una a la vez, el objetivo es entrenar el modelo  $f_T : X \rightarrow Y$  de manera que este tenga un buen desempeño en la primera tarea  $D_1$  después de haber sido entrenado en la tarea  $T$ , donde  $X \in \mathbb{R}^n$  es el espacio de entrada,  $n$  es la dimensión del vector de entrada (eje., la cantidad máxima de *tokens* de entrada en la red neuronal) y  $Y$  es el valor de la  $k$ -ésima probabilidad de inferencia.

La restricción para este modelo computacional es que no es posible acceder a los datos de las tareas anteriores, aunque se cuenta con una cantidad limitada de la información anteriormente aprendida. Un reto durante el proceso de entrenamiento del modelo con el conjunto de datos  $D_T$  es reducir o evitar el “olvido catastrófico” [16, 39, 107, 111, 112].

Un modelo de AC está compuesto por un modelo o máquina de aprendizaje (CNN, BLSTM, o un modelo pre-entrenado como Resnet). Tiene una estrategia de AC para preservar el conocimiento (pesos de la red neuronal) durante el proceso de aprendizaje de un dominio a otro. Un componente importante es el conocimiento base, el cual está asociado generalmente a los pesos de la red común o a una parte del nuevo modelo que se añade según la estrategia de AC.

El PLN es un campo de acción donde el AC puede lograr buenos resultados. La importancia del AC se asocia a que las palabras y frases tienen significados similares en diferentes dominios y tareas. Es importante tener en cuenta que las oraciones en todos los dominios siguen la misma sintaxis o gramática (según el idioma del texto). Los problemas del PLN están muy relacionados entre ellos (p.ej., reconocimiento de entidades nombradas y ABSA), lo que significa la existencia de una interconexión y afecta a cada uno de ellos de alguna forma.

Los primeros dos elementos permiten asegurar que el conocimiento aprendido puede ser usado a través de los dominios y tareas debido a que comparten la misma expresión, significado y

sintaxis.

El tercer elemento permite evidenciar que el AC puede ser empleado en diferentes tipos de tareas, como lo muestra el siguiente ejemplo: En la oración “*El restaurante tiene buena comida pero los precios son muy altos.*” se pueden extraer los atributos *comida* y *precios* de restaurante. En este ejemplo el PLN debe detectar la entidad restaurante y sus atributos, lo que muestra que ambas tareas RNE y ABSA tienen una estrecha relación. Esta situación ocurre con otras tareas del PLN [16, 96].

El uso de modelos de aprendizaje para la extracción o clasificación de opiniones en diferentes dominios, como restaurantes y hoteles, en la subtarea ABSA también puede encontrar elementos comunes. Por ejemplo, en estos dos dominios los aspectos “*precio, confort y limpieza*” son comunes [8, 43].

### 1.5.1. Definición del problema y notación

En la subtarea ABSA, dada una oración (secuencia de palabras)  $w^c = \{w_1^c, w_2^c, \dots, w_n^c\}$ , una secuencia de aspectos se define como  $w^t = \{w_1^t, w_2^t, \dots, w_m^t\}$ , y  $w^t$  es una subsecuencia de  $w^c$ .

Donde  $w^c$  y  $w^t$  pertenecen a una tarea o dominio  $D_1, D_2, \dots, D_T$ , el objetivo es entrenar el modelo  $f_T : X \rightarrow Y$  de manera que este tenga un buen desempeño en la primera tarea  $D_1$  después de haber sido entrenado en la tarea  $D_T$ , y predecir la polaridad del sentimiento  $y$  de la oración  $w^c$  hacia el aspecto objetivo  $w^t$ , donde  $y \in \{\text{positivo}, \text{negativo}, \text{neutral}\}$ .

Los datos pueden pertenecer a diferentes dominios de entrada (p.ej., opiniones sobre hoteles, efectos electrodomésticos, elecciones presidenciales) o pueden pertenecer a diferentes dominios de entrada de diferentes tareas (p.ej., análisis de sentimientos, reconocimiento de nombres de entidades, detección de tópicos) y los modelos deben predecir las clases o tareas asociadas.

Existen tres configuraciones o contextos donde evaluar los modelos de AC [2, 113, 114], estas son:

- Aprendizaje Incremental de Clases (*Class Incremental Learning*; CIL): Las tareas contienen clases que no se superponen. Solo se construye un modelo para todas las clases vistas durante el aprendizaje secuencial de las tareas. En las pruebas, no se proporciona información de la tarea. Esta configuración no es adecuada para análisis de sentimientos porque las tareas tienen por lo general las mismas clases.

## 1.5 Propuestas de modelos de aprendizaje continuo

---

- Aprendizaje Incremental de Tareas (*Task Incremental Learning*; TIL): Se construye un modelo para cada tarea en una red compartida. En las pruebas, el sistema necesita la tarea (p.ej., el dominio sobre opiniones teléfonos inteligentes) a la que pertenece esta instancia (p.ej., "La calidad del sonido es excelente") y usa solo el modelo asociado a la tarea o dominio para clasificar la instancia. Requerir la información de la tarea (por ejemplo, el dominio del teléfono) es una limitación, porque el usuario no debería tener que proporcionar esta información para una instancia de prueba o cuando se encuentre en producción el modelo computacional.
- Aprendizaje Incremental de Dominios (*Domain Incremental Learning*; DIL): En este contexto, las clases a predecir no varían para los diferentes dominios o conjuntos de datos a aprender (e.d., en ABSA usualmente las clases son positivo, neutral, negativo).

En este trabajo se aplica DIL para aprender a predecir aspectos entre los posibles valores o clases: positivo, neutral o negativo. Porque el objetivo es obtener la clasificación de la polaridad o sentimiento que expresa el aspecto (e.d., palabra) en el contexto de una oración, y no varían las clases aunque sí los dominios con los que son entrenados los modelos.

### 1.5.2. Clasificación de los modelos de aprendizaje continuo

En años recientes ha aumentado el interés en el AC y en especial su relación con modelos de AP [16, 36, 38, 115]. Las estrategias empleadas en varios trabajos se agrupan en:

- **Aislamiento de parámetros (*Parameter insolation*):** Se proponen modelos que definen subconjuntos de parámetros para cada tarea, con el objetivo de que no exista olvido una vez el modelo aprende una tarea específica [116, 117]. Un inconveniente de este enfoque es que cuando no se establecen restricciones al tamaño de la arquitectura durante el aprendizaje, pueden incrementarse nuevas ramas mientras se congelan las de las tareas anteriores. Otro enfoque costoso en memoria y tiempo es dedicar una copia del modelo para cada tarea. En algunas propuestas se realiza el enmascaramiento de partes de la arquitectura del modelo durante el entrenamiento de nuevas tareas, a nivel de parámetros [118] o de unidad (capa de red) [119]. Este enfoque tiene el inconveniente de necesitar un modelo para predecir la activación de las máscaras correspondientes o ramas de cada tarea.
- **Reproducción (*Replay*):** Aquí preservan subconjuntos de ejemplos de las tareas ante-

## 1.5 Propuestas de modelos de aprendizaje continuo

---

riores o crean seudo-ejemplos con un modelo generativo. Estos ejemplos se muestran nuevamente durante el aprendizaje del modelo actual con los objetivos de restringir la optimización de la pérdida en la nueva tarea y evitar disminuir la eficiencia en la predicción para las anteriores.

Esta estrategia tiene la limitante del gasto de memoria, al almacenar ejemplos de cada tarea o clase. Para algunas propuestas es necesario definir un método para escoger los ejemplos representativos en cada modelo o la selección aleatoria, que pudiera incurrir en una disminución de la eficiencia.

- **Regularización (Regularization)** [36]: Esta estrategia es más simple que las anteriores y reduce los requerimientos de memoria y costo computacional. Se orienta a emplear un término de regularización extra introducido en la función de costo, permitiendo penalizar grandes cambios en el conocimiento. Estos métodos se dividen en:

- *Métodos enfocados en datos*: El bloque de construcción básico en los métodos enfocados en datos son la destilación (diferenciación) del aprendizaje del modelo (entrenamiento en una tarea previa) con respecto a cuándo está siendo entrenado en los nuevos datos.
- *Métodos enfocados en la prioridad (Prior-focused methods)*: Para mitigar el olvido, los métodos enfocados previamente estiman una distribución sobre los parámetros de los modelos.

Es frecuente (p.ej., en AP) que la importancia de todos los parámetros de las redes neuronales sean asumidos como independientes. En este caso durante el entrenamiento de las tareas anteriores, los cambios de parámetros importantes son penalizados; una de las primeras propuesta fue Elastic Weigh Consolidation.

Este trabajo se centra en la estrategia de regularización, empleando *Métodos enfocados en la prioridad*, que reduce el olvido catastrófico bajo ciertas condiciones, con una cantidad limitada de recursos de neuronas; esto es una ventaja con respecto a la memoria a utilizar, pero pudiera comprometer el desempeño entre tareas anteriores y nuevas. Esta estrategia tiene buenos resultados en tareas o dominios donde la información de entrada cambia poco, tales como la inferencia de clases de animales, el color de los animales a partir de un mismo conjunto de datos [39, 120, 121], el análisis de sentimientos de dominios relativamente próximos (equipos electrodomésticos y laptops) [96].

## 1.5 Propuestas de modelos de aprendizaje continuo

---

En resumen, la selección, en esta investigación, de la regularización como estrategia para los modelos de AC tiene como motivos los siguientes:

- El costo computacional (memoria y tiempo de entrenamiento) de las estrategias **Aislamiento de parámetros y Reproducción**.
- Los modelos que usan la **Regularización** evitan almacenar entradas de datos en bruto, y reducen los requerimientos de memoria. Un término de regularización extra es introducido en la función de costo, consolidando el conocimiento previo cuando se aprenden nuevos datos.
- Se ajusta a las estrategias que se auxilian de modelos pre-entrenados, como BERT (e.d., La salida de BERT es la entrada al modelo de regularización), esto es debido a que el modelo de AC usa la salida del modelo pre-entrenado, teniendo como objetivo ajustar los valores de los parámetros de una capa de red completamente conectada (u otro tipo de arquitectura de red más compleja) asociada a la tarea de clasificación.

Para proponer un nuevo modelo que tenga mejor eficiencia que otros del estado del arte es necesario un estudio de los principales paradigmas de los modelos de AC que usan la estrategia de Regularización.

### 1.5.3. Modelos de regularización de aprendizaje continuo

Varias propuestas han estado orientadas a la creación de modelos que sigan la estrategia de Regularización. En esta sección analizaremos varias de ellas: Estrategia simple, Aprendizaje sin olvido, Consolidación elástica de pesos, Inteligencia sináptica y Estrategia de atención dura a la tarea. Para el análisis se tuvieron en cuenta aspectos tales como: su desempeño, el área de aplicación o que la forma de construcción de los modelos sea muy próxima a la requerida para resolver tareas de PLN [25, 36, 122].

La **Estrategia simple** ajusta el modelo base a través de todos los dominios de entrenamiento, sin ningún mecanismo de control del olvido catastrófico, excepto técnicas de regularización como  $L_1$ ,  $L_2$  y la regularización por omisión (*dropout*: término en inglés) [123], que pueden evitar el sobreentrenamiento y permiten obtener una mejor generalización. Esta puede provocar el olvido catastrófico si los datos de cada tarea o dominios son muy diferentes entre sí. Sin embargo, para soluciones específicas donde los datos pueden ser reentrenados (estrategias de

## 1.5 Propuestas de modelos de aprendizaje continuo

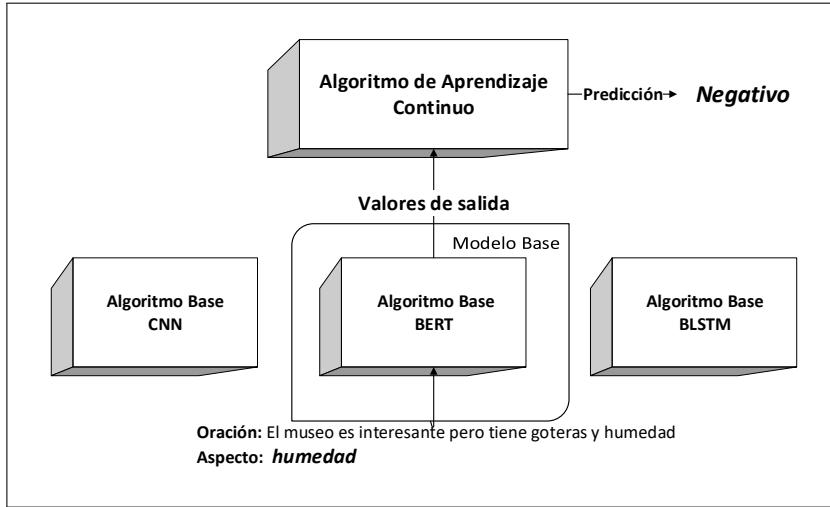


Figura 1.5: Esquema general de modelos de Aprendizaje Continuo con estrategia de Regularización.

reproducción) se pueden obtener buenos resultados de calidad [39].

Esta estrategia es empleada como modelo base para establecer valores de desempeño mínimo entre los modelos propuestos en el estado del arte [124]. Tiene un alto costo computacional y bajos resultados de calidad, porque no cuenta con métodos que optimicen el aprendizaje de los patrones o pesos entre dominios y reduzcan el tiempo y el uso de la memoria.

El **Aprendizaje sin olvido** (*Learning without forgetting*; LWF) fue propuesto en [40] inicialmente con una CNN como modelo base entre las tareas o dominios. El objetivo es forzar a que los pesos aprendidos por la red durante el aprendizaje en la tarea anterior sean similares a los obtenidos en la tarea actual, en la última capa de la red, empleando la destilación del conocimiento de la red neuronal [125].

Este modelo tiene como desventajas su dependencia a la relevancia de la tarea y que el tiempo de entrenamiento de las tareas anteriores se incrementa con la cantidad de las aprendidas [38]. No obstante, posee un bajo costo de memoria y puede emplear diferentes modelos base de AP (p.ej., CNN, LSTM, GRU, etc). Sin embargo, es dependiente de los valores en el parámetro  $\lambda_o$ , lo que requiere una evaluación adecuada del proceso de entrenamiento para encontrar los valores correctos según el problema de PLN. El LWF está relacionado con la función de costo y la de optimización del algoritmo base, pero esto no indica una estrecha relación con la preservación de los pesos de la red neuronal.

## 1.5 Propuestas de modelos de aprendizaje continuo

---

En la propuesta de [126] se emplea un modelo LWF para resolver la subtarea de extracción de aspectos en ABSA. En este trabajo se obtienen buenos valores de calidad con un F1-macro de 0.78, pero no existe una comparación con otros modelos de AC. Un análisis comparativo con otros modelos base (e.d., la propuesta usa CNN en el algoritmo base) como LSTM permitiría estimar mejor la relevancia de esta propuesta [126].

El modelo LWF tiene un costo computacional cercano a  $O(1)$ , pero la ecuación de pérdida que propone no permite compensar eficientemente la modificación de los pesos de la red neuronal durante el entrenamiento de tareas. Esto ocurre debido a que no es posible evitar grandes cambios entre los pesos de las capas iniciales y las de mayor abstracción al estimar donde ocurren grandes variaciones, y esto no permite un AC más eficiente.

El modelo de **Consolidación elástica de pesos (Elastic Weight Consolidation; EWC)** [41] consiste en una restricción cuadrática en la diferencia entre los parámetros de las tareas viejas y nueva, que hace más lento el aprendizaje de los pesos relevantes a la tarea y que fueron ajustados y aprendidos anteriormente. La relevancia del parámetro  $\theta$ -ésimo respecto a los datos de entrenamiento de la tarea  $D$  es obtenida con la distribución *a posteriori*  $p(\theta|D)$ . Asumiendo un escenario donde existen dos tareas independientes  $A$  y  $B$  con  $D_A$  y  $D_B$  el valor del logaritmo de la probabilidad *a posteriori* está dado por la regla de Bayes  $\log(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B)$ .

Donde la probabilidad *a posteriori*  $\log p(\theta|D_A)$  incluye toda la información sobre la tarea anterior. Sin embargo, dado que el término es intratable porque se necesitaría almacenar toda la información de las tareas anteriores de forma acumulativa, EWC aproxima esta a la distribución Gaussiana con la media dada por el parámetro  $\theta^*$  y los valores de la diagonal de la matriz de información de Fisher [41].

Aunque en [127] se logra para el cálculo de la matriz de información de Fisher una complejidad computacional de  $O(n^2 \lg n)$ , donde  $n$  es la cantidad de parámetros. Esto hace costoso este algoritmo, porque la matriz es creada antes del cálculo del gradiente de los pesos de la red y la propagación hacia atrás, lo que aumenta el cómputo sobre todo en modelos donde la capa de salida es de pocas dimensiones [39].

El EWC es clásico para problemas de procesamiento de imágenes y PLN, que constituye una línea inicial para evaluar nuevas propuestas [38, 119]. No existen muchas propuestas en el área del análisis de sentimientos que usen el EWC, relacionado con lo incipiente del AC en el área

## 1.5 Propuestas de modelos de aprendizaje continuo

---

del PLN [25].

La propuesta de [128] usa un modelo EWC, y es aplicada al análisis de sentimientos sobre documentos, pero puede ser aplicada al ABSA [4] debido a su cercanía a esta subtarea. Esta propuesta no se enfoca en reducir el olvido catastrófico, porque su objetivo es maximizar el rendimiento de un dominio no analizado durante el entrenamiento del modelo y no es importante el rendimiento en los dominios de origen. Por esta razón, en [128] primero se entrena con todo el conjunto de datos de los dominios anteriores, y luego evalúan el desempeño en un conjunto de datos de un dominio objetivo de prueba. Los cuatro dominios o tareas son opiniones de libros, dvd, efectos electrónicos y cocina. Todos los conjuntos tienen una cantidad balanceada de 1000 opiniones positivas y 1000 negativas.

La forma de evaluación de esta propuesta no permite que el modelo de AC conozca sobre las instancias del dominio objetivo o de prueba, y solo permite obtener buenos resultados por las coincidencias entre las instancias o ejemplos de los dominios de entrenamiento y el objetivo. Para esta propuesta se obtuvo un valor de exactitud de 0.80, teniendo como dominio de prueba u objetivo el de opiniones sobre cocina (no se conocen los resultados para otros dominios). El análisis de los resultados en [128] no permite apreciar todo el aporte de la propuesta porque solo se muestran los resultados sobre los dominios mencionados anteriormente, donde su promedio de exactitud es de 0.76.

El modelo base con mejores resultados en [128] es un CNN, con valores superiores a otras como LSTM o una arquitectura similar a BERT. Sin embargo, no se compara con otras arquitecturas como BLSTM que durante el aprendizaje de un modelo tiene en cuenta de una forma más precisa las palabras cercanas al aspecto dentro de la oración.

En [128], los mejores resultados se alcanzan con un modelo base de CNN y no con una arquitectura similar a BERT y que difiere de los resultados obtenidos por otras propuestas [25, 44]. Tampoco se inicializó el modelo base BERT con los pesos pre-entrenados y no se tienen en cuenta estos datos. Esto no permite emplear la potencia real de BERT, donde el entrenamiento con grandes conjuntos de datos genera o produce una capa de salida con valores muy cercanos a las palabras que co-ocurren en el contexto. Los autores de este trabajo sí usan para el modelo de CNN el vector pre-entrenado de Word Embedding de Glove. Estas diferencias tienen un impacto real en los valores de calidad obtenidos. En [128] no se evalúa el olvido catastrófico entre los dominios, lo que no permite estimar el aporte del modelo de AC propuesto.

## 1.5 Propuestas de modelos de aprendizaje continuo

---

El modelo **Inteligencia sináptica (Synaptic Intelligence; SI)**, presentado en [129] para problemas de clasificación de imágenes. Es una mejora al costo computacional del EWC de [41], porque no necesita la matriz de Fisher y el almacenamiento de información en ella. En la ecuación 1.2 del modelo SI, cuando aprende una nueva tarea se modifica la función de costo  $\mathcal{L}_{new}^*$  con una función de pérdida que incluye las funciones de pérdida de las tareas previas  $\mathcal{L}_o^*$ . Esto se hace con el objetivo de penalizar los cambios de las sinapsis más relevantes (pesos en las neuronas)  $\theta_i$ .

$$\mathcal{L}_{new}^* = \mathcal{L}_{new} + c \sum_i \Omega_i^n (\theta_i^* - \theta_i)^2 \quad (1.2)$$

Donde  $c$  es un parámetro para balancear el aprendizaje entre la tarea nueva y las tareas anteriores,  $\theta_i^*$  son los parámetros de la tarea nueva o actual y las tareas anteriores, y  $\Omega_i^n$  es una regulación para pesar la fuerza o influencia por parámetros. En este método, a diferencia de EWC, el cálculo de la relevancia sináptica es obtenido durante el descenso del gradiente. Esto es una ventaja de este modelo porque su costo computacional está muy cercano al modelo de AP base que se usa al aprender en los diferentes dominios.

A pesar de que no se encontraron trabajos que hagan uso de SI para PLN o ABSA, en esta investigación se ha considerado su aplicación teniendo en cuenta la optimización que este hace sobre el costo computacional de EWC, además de las ventajas de estimación de la relevancia de los pesos de las neuronas durante el descenso del gradiente.

La **Estrategia de atención dura a la tarea (Approach Hard Attention to the Task; HAT)** presentada en el trabajo de [119] para enfrentar el problema de clasificación de imágenes, está basado en un MA que mantiene la información de las tareas previas concurrentemente al proceso de aprendizaje de la tarea actual. Para esto se cuenta con vectores de atención a través de compuertas embebidas de cada tarea y se apoya, al igual que SI, en el mecanismo de descenso del gradiente. Los vectores de atención de las tareas previas son usados para definir una máscara y restringen las actualizaciones de los pesos de las redes neuronales en la tarea actual. Las máscaras permiten que una parte de los pesos permanezca sin ser modificada mientras que el resto se adapta a la nueva tarea.

El HAT supera a los modelos EWC, LWF y SI durante la evaluación de su desempeño en [119] para un tarea de clasificación de imágenes con el pre-modelo de AP *Alexnet* [130].

## 1.5 Propuestas de modelos de aprendizaje continuo

---

En [131] se propone un modelo orientado a la clasificación de sentimientos a nivel de documentos, y está estrechamente relacionado con el modelo HAT. A pesar de los pocos trabajos sobre AC para ABSA, se analizó la posibilidad de hacer uso de sus principales conceptos. En esta propuesta, al igual que ocurre con HAT, se entrena una máscara binaria usando la atención dura. Sin embargo, en el caso de HAT es usada para identificar qué conocimientos aprendidos en las tareas anteriores deben ser protegidos para que la nueva tarea no modifique este conocimiento previo, y es efectivo para evitar el olvido catastrófico. El modelo KAN tiene dos redes neuronales, la de AC Principal o *Main Continual Learning* (MCL) y la red de accesibilidad (RA), mientras que HAT solamente define una red neuronal, por lo que es más costosa en memoria y tiempo de entrenamiento.

La red MCL almacena el conocimiento y aplica las máscaras de AC al conocimiento base. Esta configuración permite a KAN no solamente adaptar el conocimiento compartido a través de tareas para producir modelos más eficaces, sino también evitar el olvido catastrófico. KAN alcanza una exactitud de 0.85 y realiza una amplia evaluación con otros modelos del estado del arte del AC como EWC, HAT y otros trabajos de AC para el análisis de sentimientos basado en documentos, como el propuesto en [132], superándolos a todos.

Sin embargo, se debe destacar como desventaja el alto costo computacional y de memoria que posee este modelo y que se establece debido a que las redes neuronales de AC y MCL están constituidas por un modelo que sigue una arquitectura GRU; el costo computacional de cada capa de la red GRU es de  $O(r \cdot d^2)$ , donde  $r$  es el tamaño de la secuencia de vectores de entrada y  $d$  la dimensión de cada vector de entrada a la red. Para cada tarea se produce una matriz binaria, asociada a los documentos presentes en el conjunto de datos.

El método KAN tampoco evalúa la entrada de las redes con otros modelos pre-entrenados como BERT, a pesar de que es considerado un estándar para comparar los modelos en PLN, y solo se emplea un vector pre-entrenado de Word Embedding de Glove [8, 25].

Aunque los conjuntos de datos para el entrenamiento están balanceados y se usa una gran cantidad (24), en KAN no se indica si los resultados tienen en cuenta todos los posibles órdenes de las tareas durante el AC. El orden de aparición de las tareas pudiera afectar el desempeño del modelo, por la distribución de los datos en cada uno de los conjuntos [36].

Un resultado a considerar, presentado en [131], es la forma de evaluar el olvido catastrófico. Para esto, al terminar de entrenar la última tarea, se estima la calidad de la clasificación para

## **1.6 Consideraciones finales del capítulo**

---

cada una de las tareas anteriores, empleando el modelo entrenado en la tarea actual. Para el problema de la estimación del olvido catastrófico no existe una medida estándar [25, 36], pero la medida presentada en [131] tiene una estrecha relación con este problema, por la forma de tener en cuenta los resultados entre las tareas anteriores y la calidad de la clasificación.

### **1.6. Consideraciones finales del capítulo**

El uso de los modelos de aprendizaje profundo evitan la ingeniería de características y disminuyen los costos en la creación de sistemas para lograr el análisis de sentimientos basado en aspectos, siendo LSTM y los mecanismos de atención los modelos más empleados.

Las propuestas de modelos que usan CNN no son muy empleadas para problemas de PLN, por los investigadores, debido a que esta estrategia se orienta a resolver problemas de procesamiento y clasificación de imágenes (por las características de la arquitectura de los modelos que la emplean). La adaptación de CNN a problemas de PLN es posible pero incurren en el aumento del costo computacional (e.d., en memoria) al necesitar establecer un tamaño máximo para el texto presente en los ejemplos de los conjuntos de datos.

Los modelos de representación textual más exitosos han sido los Word Embedding, siendo BERT el de mejor desempeño en análisis de sentimientos basado en aspectos, debido a que la arquitectura del modelo y los conjuntos de entrenamientos empleados para el aprendizaje permiten conservar la relación semántica entre las palabras de un texto.

Los modelos de aprendizaje profundo analizados obtienen valores de calidad mayores que 0.80 de exactitud, al aplicarlos a dominios o conjunto de datos por separado. Sin embargo, al ser aplicados a varios dominios durante la fase de entrenamiento, la efectividad disminuye por el olvido o modificación de los parámetros del modelo.

La estrategia de regularización en el aprendizaje continuo es más eficiente en cuanto al uso de memoria y tiempo de entrenamiento de modelos, que otras en el estado del arte, y es la seleccionada para guiar el diseño e implementación de los modelos propuestos.

Encontrar un modelo que permita el aprendizaje continuo sin perder efectividad en la calidad del proceso de aprendizaje iterativo, constituye un reto de investigación y permitiría la creación de modelos y sistemas más eficaces en ABSA. En este sentido, los capítulos siguientes proponen varias soluciones al reto planteado.

## CAPÍTULO 2

# Propuesta de modelos computacionales para el análisis de sentimientos basado en aspectos

En este capítulo se describen las características de un nuevo modelo que combina el aprendizaje continuo y profundo para la extracción de aspectos en ABSA [4], que es necesario para la identificación de los aspectos en una oración. Después se relacionan las características de un modelo de aprendizaje continuo y profundo para la clasificación de aspectos en ABSA, importante para definir la polaridad (e.d., positivo, negativo o neutral) del aspecto. Finalmente, se muestra la evaluación del modelo con respecto a varias propuestas del estado del arte, comparándolo con medidas clásicas del desempeño, en cuanto a la calidad de la clasificación y el olvido catastrófico.

### 2.1. Propuesta de modelo computacional para la extracción de aspectos

En esta sección se propone un nuevo modelo para la extracción de aspectos, basado en la combinación de una red neuronal convolucional (CNN) y un modelo de AC. La principal contribución de esta propuesta es la creación de un marco de trabajo de AC para la subtarea de extracción de aspectos en ABSA que es realizada en algunas propuestas empleando modelos de CNN [34, 133], donde se obtienen las características de posibles *n*-gramas para crear la representación semántica latente de las oraciones [18]. En la investigación presentada en [133] se muestran buenos resultados en la extracción de aspectos, obteniéndose un desempeño de F1-macro igual a 0.86. Los resultados mostrados en [133] se derivan del entrenamiento del modelo en un solo dominio (e.d., dominios de opiniones acerca de restaurantes o sobre laptops).

El nuevo modelo propuesto en esta investigación está inspirado en [133], pero se diferencia de este en que es entrenado y evaluado en un entorno de AC para diferentes dominios. Es importante definir las partes del nuevo modelo para una mejor comprensión de su desempeño.

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

### **2.1.1. Descripción del modelo para la extracción de aspectos en múltiples dominios**

El nuevo modelo para la extracción de aspectos ha sido diseñado para un entorno donde las clases a predecir no varían en diferentes dominios o conjuntos de datos (*Domain Incremental Learning*; DIL) [36]. La arquitectura de AP empleada está basada en la combinación de una arquitectura CNN, propuesta en [133], y un componente para el AC que sigue las pautas de una estrategia de regularización nombrada “Aprendizaje sin olvido” (*Learning without forgetting*; LWF) propuesto en [40], como se muestra en la figura 2.2.

La propuesta presentada en esta investigación está inspirada en esta estrategia por su éxito al reducir el olvido catastrófico durante el aprendizaje de modelos CNN. LWF fue aplicada a la clasificación de imágenes y en un entorno en el que para un mismo conjunto de datos en cada nueva tarea se aprenden nuevas clases (e.d., imágenes de perros y gatos, mamíferos o peces, etc.). En nuestra propuesta, el modelo LWF fue adaptado al PLN y al entorno DIL de AC para etiquetar palabras en oraciones como aspectos o no.

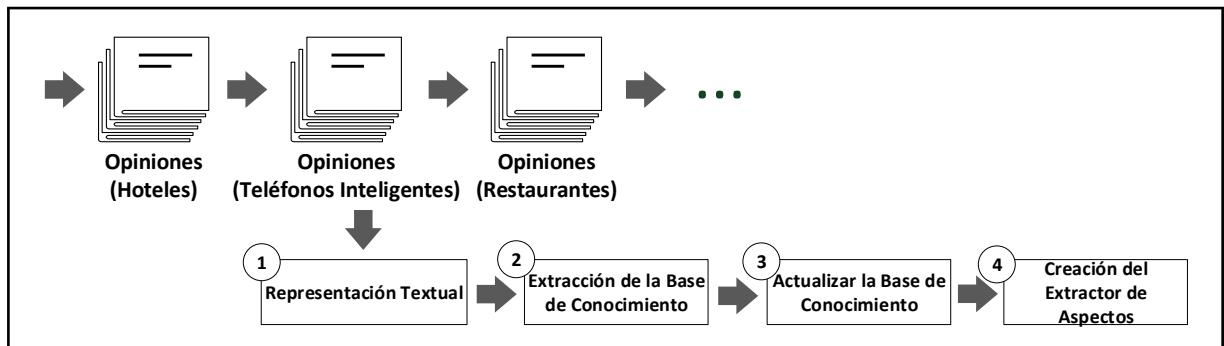


Figura 2.1: Etapas para la creación de un modelo para la extracción de aspectos basado en el aprendizaje profundo y continuo.

El modelo propuesto está compuesto por cuatro etapas principales, como se muestra en la figura 2.1. Para cada dominio, en el modelo de AC, una capa de entrada está conectada a la última capa del CNN. El objetivo del modelo base CNN es aprender las características dependientes de cada domino y almacenar esta información en sus parámetros (e.d., pesos de las neuronas).

- **Etapa 1. Representación textual:** Inicialmente se aprende un conjunto de aspectos a

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

partir de un conjunto de datos no etiquetados, mediante el uso de reglas lingüísticas (e.d., se crea una herramienta auxiliar al modelo computacional; para los casos en que no se detecte una palabra como aspecto), este proceso se realiza en paralelo al proceso de entrenamiento del modelo para la extracción de aspectos. En el Anexo C se muestra un diagrama del proceso de entrenamiento del modelo.

Esta etapa recibe las opiniones textuales originales y retorna el vector de la capa de salida del pre-modelo BERT [31] para cada oración, a través de los siguientes pasos: (i) pre-procesamiento textual de las opiniones, aplicando un divisor de oraciones y un etiquetador morfológico, y (ii) obtener el modelo de vector de palabras a partir de un pre-modelo BERT.

- **Etapa 2: Extracción del conocimiento básico:** Se aprende el conocimiento a incluir en la base de conocimientos (BC) dependiendo del proceso de entrenamiento del CNN para cada dominio. En la salida de esta etapa se obtienen los nuevos parámetros y se retorna el vector asociado a la última capa del CNN.
- **Etapa 3: Actualización de la base de conocimientos:** Se evita o reduce el olvido catastrófico a través del proceso de entrenamiento del modelo de aprendizaje continuo LWF.

El costo (e.d., el error de predicción más el factor de regularización) que se obtiene en el modelo durante su entrenamiento en el dominio actual es comparado con los resultados en los dominios anteriores. La BC es enriquecida por el proceso de entrenamiento del modelo CNN (p.ej., la información de los aspectos en cada nuevo dominio) y el ajuste de los pesos de la red neuronal a partir de la regularización propuesta en [40].

- **Etapa 4: Creación del extractor de aspectos:** Se hace posible que el modelo de AP pueda resolver la subtarea de extracción de aspectos en múltiples dominios. La configuración final del CNN y del modelo de AC se obtiene de los parámetros comunes en la BC.

En la figura 2.2 se muestran los diferentes elementos del modelo propuesto.

## 2.1 Propuesta de modelo computacional para la extracción de aspectos

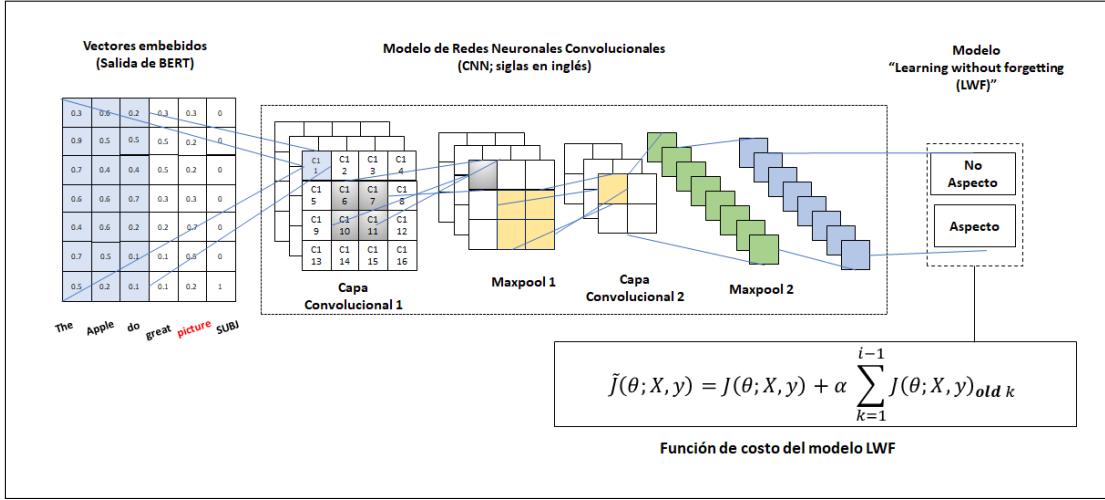


Figura 2.2: Esquema general del modelo para la extracción de aspectos Lwf-CNN-kgR.

En la figura anterior se muestra cómo el modelo base (CNN) entrega su salida a una capa completamente conectada asociada a los valores de clasificación. En esta capa se crea el modelo de AC a partir de una estrategia de regularización.

Las siete capas de la arquitectura CNN usada en la propuesta presentada en esta investigación son las mismas del modelo presentado en [133].

Luego del pre-procesamiento de las oraciones, se añade, para cada palabra de la oración, su etiqueta morfológica correspondiente (p.ej., SUB para sustantivos, VERB para verbos) al vector embebido de BERT para cada *token*.

Se emplean seis etiquetas básicas (sustantivos, verbos, adjetivos, adverbios, preposiciones y otras) para codificar un vector binario de seis dimensiones. Este vector es concatenado al vector de salida obtenido del modelo pre-entrenado BERT, a partir de una secuencia de *tokens* de una oración [31].

Esto significa que cada oración a entrenar tiene como entrada al modelo CNN un vector de tokens; cada token es un vector de 88 dimensiones (e.d., 82 es el tamaño máximo de *tokens* de las oraciones en los conjuntos de datos, más seis dimensiones asociadas al vector de la etiqueta morfológica de la palabra).

La segunda capa de la arquitectura CNN es una capa convolucional con 100 características como mapa y se usa un filtro igual a dos. La salida fue calculada usando una tangente hiperbólica.

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

La tercera y quinta capa son capas maxpooling (e.d., capas donde se realiza una operación de agrupación que calcula el valor máximo en partes del vector de entrada), con un *pool* igual a dos. La cuarta capa es una capa convolucional, con 50 características como mapa con un filtro igual a tres. La salida de esta capa fue calculada con la función de la tangente hiperbólica. La sexta capa está completamente conectada. El paso o salto (*stride*: término en inglés) en la ejecución de la convolución entre la capa anterior y la capa convolucional es uno, para considerar la relación entre cada palabra [133].

Se respetaron los valores seleccionados en la propuesta de [133], para los mapas de características de la segunda (100) y cuarta capa (50) de la red CNN.

Se emplea la regularización por omisión (*dropout*: término en inglés) en la penúltima capa y con una restricción de tipo  $L_2$  para calcular los pesos del vector y aumentar la variabilidad de las posibles soluciones [17, 18]. La función de convolución de la capa de entrada del modelo CNN durante el proceso de aprendizaje, tiene en cuenta una ventana de cinco palabras con respecto a la palabra que se analiza (p.ej., la palabra analizada y dos palabras a la derecha y a la izquierda). Esta estrategia responde a la posible relación entre la palabra asociada al aspecto y las palabras más cercanas a este. Otros valores para el tamaño de la ventana fueron evaluados, pero no arrojaron resultados que incrementaran la exactitud del modelo. La estimación del error fue hecha aplicando el algoritmo de Viterbi [134] a la capa de salida de cada dominio.

Los modelos computacionales de AP necesitan gran cantidad de información de datos etiquetados para el entrenamiento inicial [18]. Sin embargo, para ABSA estos datos etiquetados son escasos. El uso de pre-modelos como BERT en ABSA (y en otras tareas del PLN), que han sido entrenados en grandes conjuntos de datos de dominios o de textos digitales en idiomas como el inglés, permite reducir esta desventaja [31, 135]. Esto se debe a que la arquitectura de red neuronal en BERT permite almacenar en los pesos de la red neuronal aprendidos durante el entrenamiento mucha información asociada al contexto o vecindad de una palabra en diferentes textos digitales.

Las posibles variantes de aparición de conjunto de palabras o *tokens* en las oraciones u documentos que conforman las instancias de los conjuntos de datos de la tarea ABSA, tienen una alta probabilidad de haber sido tenidos en cuenta durante el entrenamiento del modelo BERT<sup>1</sup>.

---

<sup>1</sup>El modelo BERT en su entrenamiento tuvo como conjuntos de datos el BookCorpus (800 millones de palabras) y la Wikipedia en idioma inglés (2500 millones de palabras). Para el caso de Wikipedia se tomó en cuenta solo el texto, rechazando listas, tablas y encabezados.

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

Es por esto que BERT puede emplearse como capa de entrada o modelo base de varias tareas de PLN, en especial de ABSA, teniendo en cuenta una capa final de clasificación que recibe como entrada la salida de la última capa de BERT.

El nuevo modelo usará una capa de red o una estructura neuronal más compleja, teniendo como entrada el vector antes descrito. Esta estrategia también permite reajustar los parámetros de BERT a la tarea específica de PLN (p.ej., ABSA), con el objetivo de reconocer los patrones del PLN presentes en ella.

Otra forma de reducir la desventaja o problema de contar con pocos ejemplos para el entrenamiento de un modelo AP se realiza en la propuesta de [133]. En esta se utiliza una estrategia de etiquetado de aspectos que expresan sentimientos, mediante el uso de reglas lingüísticas, que identifican posibles aspectos en una oración, usando la relación sintáctica entre las palabras y si la palabra se encuentra presente en el recurso SenticNet<sup>2</sup>. La relación sintáctica entre palabras es determinada con la herramienta para el PLN (módulos y marco de trabajo para el desarrollo de aplicaciones) spaCy<sup>3</sup>.

El modelo propuesto para la extracción de aspectos es nombrado aprendizaje sin olvido con reglas lingüísticas (*Learning without Forgetting with Linguistic Rules; Lwf-CNN-lgR*).

Este modelo emplea como estrategia de regularización para el AC el modelo LWF con un menor costo computacional que otros como EWC y los valores del parámetro  $\lambda_o$  permiten a los investigadores determinar la influencia del aprendizaje de dominios anteriores en el dominio actual.

El LWF ha sido frecuentemente empleado para la clasificación de imágenes [39, 136], pero fue adaptado a un problema de PLN en esta investigación; tiene como principales diferencias a la propuesta original:

- Se combina una arquitectura CNN adaptada al PLN con el modelo de AC nombrado LWF en ABSA.
- Se modifica el entorno de AC para entrenar un modelo en la clasificación de tres clases con información o ejemplos de diferentes conjuntos de datos (p.ej., en la versión original se emplea en un contexto TIL).

---

<sup>2</sup><https://senticnet.net/>

<sup>3</sup><https://spacy.io>

## 2.1 Propuesta de modelo computacional para la extracción de aspectos

---

- La capa de clasificación, en la propuesta original, es una red neuronal de dos capas con 4096 nodos en la capa oculta. En la propuesta de esta investigación es una sola capa de dos neuronas.
- El vector de entrada al modelo base es la salida de un modelo pre-entrenado BERT.
- Se crea una herramienta auxiliar al modelo computacional mediante el uso de reglas lingüísticas.

Para poder evaluar el desempeño de la nueva propuesta en la tarea ABSA, se mantuvieron tanto la función de costo de LWF como su principio de funcionamiento.

Una desventaja del LWF es que mantiene en memoria, durante el entrenamiento, las capas de clasificación para los dominios anteriores. Esto se hace con el objetivo de estimar la pérdida o error de una instancia del conjunto de entrenamiento del dominio actual con respecto al modelo obteniendo en cada uno de los dominios anteriores. La sumatoria de estos valores forman parte del término de regularización de la función de costo propuesta en LWF para penalizar el resultado de la función y evitar grandes cambios durante el cálculo del DGE.

La función de costo en modelos de máquinas de aprendizaje se descompone frecuentemente como una suma sobre los ejemplos de entrenamiento de la función de pérdida [19, 51]:

$$J(\theta) = \mathbb{E}_{f(x, \theta) \sim \hat{p}_{data}} L(x, y, \theta) = \frac{1}{n} \sum_{i=1}^n L(x_i, y_i, \theta) \quad (2.1)$$

Donde  $\theta$  define al conjunto de parámetros en la función  $f(x; \theta)$  que es optimizada por la red neuronal de un modelo de AP y  $n$  la cantidad de ejemplos del conjunto de entrenamiento,  $x$  representa el vector o ejemplos a entrenar y  $y$  su etiqueta o valor a inferir.<sup>4</sup>

En la ecuación 2.1,  $n$  representa la cantidad de elementos del conjunto de datos de entrenamiento y  $L(x, y, \theta)$  es la función de pérdida para cada ejemplo o instancia durante el entrenamiento. Estando  $L(x, y, \theta)$  relacionada a  $f(x; \theta)$  porque indica cuánto se acerca la predicción al valor o etiqueta de clasificación.

Para una distribución condicional  $p(y|x; \theta) = f(y|x; \theta)$ , el principio de máxima verosimilitud

---

<sup>4</sup>La letra J se refiere a la matriz Jacobiana. Esta matriz contiene las derivadas parciales asociadas al vector de entrada de la red neuronal. Dada la función  $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ , entonces la matriz Jacobiana  $J \in \mathbb{R}^{n*m}$  de  $f$  se define por  $J_{i,j} = \frac{\partial}{\partial x_j} f(x)_i$  [19].

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

condicional [137] propone el uso de  $-\log(y|x; \theta)$  como función de pérdida<sup>5</sup>. En el caso de los modelos de AP esta función es la que se utiliza usualmente [19, 42]. Teniendo como referencia la ecuación 2.1, en esta investigación la función de pérdida para los modelos se define como::

$$L(X, y, \theta) = -\log(y|X; \theta) \quad (2.2)$$

Un reto importante para los modelos de máquinas de aprendizaje es lograr un buen desempeño, ajustado a los datos de entrenamiento (e.d., al problema o tarea que enfrentan), y a nuevos datos de entrada. Muchas estrategias son empleadas para tratar de reducir el error durante las pruebas del modelo, son conocidas colectivamente como regularización [19]. En esta investigación se toma como definición de regularización la propuesta en [19]: “cualquier modificación que se haga a un algoritmo de aprendizaje que pretenda reducir el error de generalización pero no el error de entrenamiento”.

Varias de las estrategias de regularización usadas para las redes neuronales se basan en limitar la capacidad del modelo, esto se hace agregando una función de penalización a los parámetros de la red neuronal  $\Omega(\theta)$  a la función de costo  $J$ , trasformando la función de costo general de la ecuación 2.1 en:

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta) \quad (2.3)$$

Donde  $\alpha \in [0, \infty)$  es un hiperparámetro que pondera la contribución relativa de la función de penalización  $\Omega$  con respecto a la función de costo estándar  $J$ .

En la ecuación 2.4 se muestra la función de costo del modelo LWF, similar a la función de costo utilizada para entrenar la red neuronal de un modelo AP y combina una función de pérdida (como se representa en la ecuación 2.3) con una función de regularización:

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha \sum_{k=1}^{m-1} J(\theta; X, y)_{oldk} \quad (2.4)$$

Donde  $\alpha^6$  pondera la contribución de la función de penalización y  $\Omega(\theta)$  (en la ecuación de la

---

<sup>5</sup>El uso de esta función permite que sea computacionalmente más fácil encontrar las derivadas parciales y reducir  $f(x)$  al mover  $x$  en pequeños pasos con el opuesto de la derivada.

<sup>6</sup>El valor escalar  $\alpha$  es  $\lambda_o$  en la función original propuesta en [40] y permite penalizar a la influencia del

## 2.1 Propuesta de modelo computacional para la extracción de aspectos

---

función de costo 2.3) se define por:

$$\Omega(\theta) = \sum_{k=1}^{m-1} J(\theta; X, y)_{oldk} \quad (2.5)$$

En 2.5 optimizan (regularización) los parámetros del modelo de AC (e.d., pesos de la capa de neuronas asociadas a la clasificación de aspectos y los parámetros comunes en el modelo base  $\theta_s$ ). El objetivo es imponer restricciones a la modificación de los parámetros en  $\theta_s$ , cuando se pasa de un conjunto de datos (domino) a otro durante el entrenamiento, para evitar el olvido catastrófico.

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha(\sum_{k=1}^{m-1} J(\theta; X, y)_{oldk}) \quad (2.6)$$

$$\tilde{J}(\theta_n; X, y) = -\log(y|X; \theta) + \alpha(\sum_{k=1}^{m-1} -\log(y|X; \theta)_{oldk}) \quad (2.7)$$

Donde  $\sum_{k=1}^{n-1} -\log(y|X; \theta)_{oldk}$  es la suma de las pérdidas en la capa de clasificación asociada a los dominios anteriores,  $m$  representa la cantidad de conjuntos de entrenamiento y  $k$  representa la capa de clasificación del modelo anterior de la que se obtiene el valor de inferencia para el vector de salida del modelo base. En la ecuación 2.7,  $\alpha$  es parte de la regularización de la función de costo y se usa para lograr un balance o penalización entre la tareas anteriores y la actual. Durante la propagación hacia atrás del descenso del gradiente se aplica una penalización a los pesos de la red conocida por *decadencia del peso* (e.d., *weight decay*) con valor 0.0005 (respetando el valor definido en [40]; propuesta inicial de LWF). Se experimentaron con otros valores (p.ej: 0.0001, 0.002) menores con el objetivo de encontrar mejores soluciones en el espacio de búsqueda, sin saltos muy grandes durante el DGE [51], pero no mostraron una eficacia significativa.

El Lwf-CNN-lgR se muestra en el algoritmo 1. Su objetivo consiste en aprender a clasificar las palabras etiquetadas como aspecto en cada oración perteneciente a los conjuntos de datos de cada dominio. Los conjuntos de datos de entrenamiento son presentados al algoritmo de forma secuencial y al finalizar el entrenamiento de cada dominio, los valores de los pesos de las neuronas de la capa de clasificación son preservados.

---

aprendizaje de dominios anteriores en el dominio actual.

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

Durante el entrenamiento, los pesos de la última capa, relacionada con la clasificación, no son inicializados a 0 u otro tipo de estrategia (p.ej., usar valores de una distribución uniforme) al pasar de un dominio a otro; para la modificación del valor, la función de pérdida realiza una compensación entre el valor por asignar al peso de la neurona y los valores de los pesos de esta neurona al finalizar el entrenamiento de los dominios anteriores. Esta estrategia permite que no se olvide completamente la información aprendida durante el proceso de entrenamiento.

La entrada del algoritmo es el token de la palabra a clasificar como aspecto o no, y los tokens de la oración a la que pertenece esta palabra en el conjunto de entrenamiento.

Los tokens de la oración son usados como conjunto de entrada del pre-modelo BERT y convertidos por este en un vector de valores reales de 82 dimensiones.

El token de la palabra a clasificar como aspecto es representado por un vector binario de seis dimensiones asociadas a su etiqueta morfológica (e.d 1 si la palabra en la oración representa un sustantivo, adjetivo, verbo, adverbio, pronombre u otro y 0 en el resto de los casos). Este vector binario se crea con el objetivo de representar las características morfológicas del posible aspecto en la oración, y obtener una representación computacional para ser procesada por un algoritmo de AP.

Ambos, el vector de valores reales y el vector binario son concatenados, construyendo un vector final de 88 dimensiones como se describió anteriormente, que junto a la etiqueta de clasificación (p.ej, aspecto o no aspecto) es la entrada al algoritmo 1.

La salida de este algoritmo es el conjunto de los valores calculados para los pesos de las neuronas de la arquitectura del modelo base CNN y de la capa de clasificación del modelo LwF.

En el algoritmo 1 se define el subíndice  $s$  que indica los parámetros compartidos por todos los dominios (modelo CNN), el subíndice  $c$  está relacionado con parámetros específicos del dominio actual, y el subíndice  $p$  está asociado con los parámetros específicos del último dominio.

En el algoritmo 1, los parámetros que corresponden a la capa de salida del dominio actual son aleatoriamente inicializados, como se muestra en la línea 1. Los parámetros de cada capa de salida de los dominios previos son almacenados para ser usados en la función de pérdida del modelo. El modelo CNN es entrenado para cada oración del conjunto de datos de cada dominio. Las salidas son obtenidas para los dominios anteriores y el dominio actual, como

## 2.1 Propuesta de modelo computacional para la extracción de aspectos

---

**Algorithm 1** Algoritmo de aprendizaje continuo para la extracción de aspectos

---

**Entrada :**

$\Theta_s$	▷ Parámetros compartidos del modelo CNN
$\Theta_p$	▷ Parámetros de los dominios anteriores (capa de clasificación)
$X_c, Y_c$	▷ Oraciones y aspectos etiquetados en el dominio $c$ actual

**Salida :**

$\Theta_s$	▷ Modelo CNN
$\Theta_c$	▷ La capa de salida del último dominio
1: $\Theta_c = \text{RandInit}()$	▷ Nuevos parámetros inicializados aleatoriamente
2: <b>for</b> $x_i, y_i \leftarrow X_c, Y_c$ <b>do</b>	
3: $Y_p = \text{TrainCNN}(x_i, \Theta_s, \Theta_p)$	▷ Salida de los dominios anteriores
4: $y_c = \text{TrainCNN}(x_i, \Theta_s, \Theta_c)$	▷ Salida del dominio actual
5: $Loss'_i = \lambda_o \sum_{k=1}^{i-1} Loss_{old}^k(Y_p^k, y_i) + Loss_i(y_c, y_i)$	
6: <b>end for</b>	

---

muestran las líneas 3 y 4.

La línea 5 muestra cómo son minimizados los valores entre el dominio actual y los anteriores. Este proceso actualiza los parámetros de la red neuronal a través de la regularización y el descenso del gradiente. El balance que se trata de obtener entre los resultados del dominio anterior y el actual contribuyen a reducir o aliviar el olvido catastrófico, porque evita grandes cambios en el valor de los pesos de la arquitectura de red durante el cálculo y ajuste del gradiente. Un valor alto de la constante  $\lambda_o$  determina una mayor influencia de los valores de la función de pérdida evaluada en la capa de clasificación de los dominios anteriores. La capa de salida contiene el conocimiento común entre todas las capas de salida. Estos dos resultados son el contenido de la base de conocimientos.

El modelo propuesto requiere el uso de herramientas que permitan obtener la estructura grammatical del texto (p.ej., oraciones, palabras o *tokens*, etiquetas morfológicas asociadas a un *token*) y por esta razón se emplea el marco de trabajo spaCy para el PLN. El uso práctico de este marco de trabajo se valida durante la primera etapa de construcción del modelo.

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

### **2.1.2. Análisis de la complejidad computacional del entrenamiento del modelo para la extracción de aspectos**

Dada la función de costo representada en la ecuación 2.6, el DGE requiere calcular:

$$\nabla_{\theta} J(\theta) = \frac{1}{n} \sum_{i=1}^n \nabla_{\theta} L(X^{(i)}, y^{(i)}, \theta) \quad (2.8)$$

El costo computacional de esta operación es  $O(n)$  (al aumentar en grandes cantidades los ejemplos en el conjunto de datos de entrenamiento, el costo computacional en tiempo, que toma un solo ciclo del descenso del gradiente, aumenta considerablemente).

La idea del descenso del gradiente estocástico es que el gradiente es una expectativa y se puede realizar una aproximación empleando un pequeño conjunto de muestras [19].

En el caso del entrenamiento de un modelo de AP, para cada paso del algoritmo, sobre un conjunto de entrenamiento, es posible dividirlo en lotes de ejemplos  $\mathbb{B} = \{x^{(1)}, x^{(2)}, \dots, x^{(b)}\}$ , extraídos uniformemente del conjunto de entrenamiento. El valor de  $b$  se mantiene fijo como la cantidad  $n$  de ejemplos del conjunto de entrenamiento.

La sumatoria de la función de costo y el factor de regularización para cada uno de los ejemplos del lote, es promediado como se muestra en:

$$g = \frac{1}{b} \nabla_{\theta} \sum_{i=1}^b L(X^{(i)}, y^{(i)}, \theta) \quad (2.9)$$

Esta operación se realiza para lograr una aproximación que reduzca la complejidad computacional y permita converger, de una manera más rápida, a una posible solución. El empleo de lotes del conjunto de datos de entrenamiento y el promedio de sus resultados a partir de la función de costo, permiten estimar la complejidad computacional del entrenamiento de un modelo de AP para un ciclo sobre el conjunto de entrenamiento en  $O(\frac{n}{b})$ .

En el AP la cantidad de ciclos de entrenamiento  $p$  define un costo sobre el cómputo, porque establece la cantidad de veces que debe entrenarse un modelo sobre el mismo conjunto de datos para encontrar la mejor solución. Es importante considerar este valor al estimar la complejidad computacional del modelo; estableciendo  $O(p \frac{n}{b})$ .

En esta investigación se estudia el AC y se proponen modelos computacionales, que combinan-

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

do modelos de AP, realizan los  $p$  ciclos para una  $m$  cantidad de conjuntos de entrenamiento. El valor de  $m$  tiene una influencia notable en la complejidad del modelo de AC, porque cada modelo debe repetir  $m$  veces los  $p$  ciclos de entrenamiento y trasforma la ecuación del costo computacional en  $O(mp\frac{n}{b})$ .

Un análisis del costo o complejidad computacional para un modelo de AC debe tener en cuenta la influencia o costo de un modelo de AP, cuando este es su modelo base (como se propone en esta investigación). Esto se debe a que generan una cantidad importante de operaciones durante el aprendizaje de los pesos de las neuronas que conforman su arquitectura, y que pueden ser de una cantidad considerable (p.ej., el modelo BERT empleado en esta investigación tiene unos 110 millones de neuronas).

El Lwf-CNN-lgR combina el uso de un CNN como modelo base (e.d., ajusta los pesos de la red neuronal del modelo CNN durante el entrenamiento en cada dominio) y tiene un modelo para el AC de una capa completamente conectada a la salida del modelo CNN. La capa del modelo de AC contiene dos neuronas asociadas a los posibles valores a clasificar (e.d., aspecto o no).

En un modelo CNN, la complejidad computacional por capas es de  $O(krd^2)$ , según se define en [82]. Donde  $r$  es el tamaño de la secuencia o vector de entrada a cada capa,  $d$  es la dimensión del vector y  $k$  es el tamaño del kernel de las convoluciones.

El costo computacional de una capa de neuronas completamente conectada es de  $O(n^2)$ . La capa de clasificación del modelo de AC para Lwf-CNN-lg tiene una entrada de valor tres, por lo que su costo es  $O(3^2) = O(9)$ .

El análisis de la complejidad del Lwf-CNN-lgR debe comprender el costo para aprender nuevos patrones de reglas lingüísticas durante la detección de aspectos, que es parte de este modelo. Al analizar su costo computacional fue considerado como  $O(n)$ , porque se realiza sobre un único conjunto de entrenamiento. El aprendizaje de reglas lingüísticas se realiza de forma paralela al AC, sobre un solo dominio no etiquetado, y requiere la búsqueda sobre una tabla o listas de seis reglas previamente definidas.

El costo computacional del aprendizaje del modelo base del Lwf-CNN-lgR es de  $6 * O(krd^2) + O(9)$  (seis es la cantidad de capas del modelo CNN). Al reducir esta expresión a su término más general, tenemos la expresión  $O(krd^2)$ . Para este análisis no se tiene en cuenta el costo computacional del entrenamiento o ajuste del pre-modelo BERT, porque solamente es emplea-

## **2.1 Propuesta de modelo computacional para la extracción de aspectos**

---

do como generador del vector de entrada (e.d., vector de word embeddings) al modelo base CNN.

La complejidad computacional total del modelo Lwf-CNN-lgR se define por:

$$O(mp \frac{n}{b} (krd^2)) \quad (2.10)$$

Donde se define una cantidad  $m > 2$  de dominios a entrenar (se impone esta restricción para evitar un aprendizaje por transferencia). El modelo CRF con el que es comparado el Lwf-CNN-lgR, tiene una complejidad computacional de  $O(cn^2)$ , donde  $n$  es la cantidad de ejemplos o instancias del conjunto de datos y  $c$  la cantidad de clases (ver un análisis más profundo en [138]). Teniendo en cuenta un esquema de AC para el modelo CRF (como el que se realiza en esta investigación), entonces se debe tener en cuenta los  $m$  conjuntos de datos y se modifica la expresión anterior por  $O(mcn^2)$ . Durante la comparación del modelo CRF, en esta investigación no se realizaron varios ciclos sobre los conjuntos de entrenamiento porque el uso de esta técnica no se establece en la definición de este modelo.

Como se describe en 2.11,  $c \geq 2$  y  $b < n$  para el entrenamiento de un modelo de clasificación, el Lwf-CNN-lgR tiene una complejidad menor que el CRF:

$$O(mcn^2) \leq O(mp \frac{n}{b}) \quad (2.11)$$

Aunque el uso del CNN como modelo base en Lwf-CNN-lgR aumenta el costo por el uso de memoria y su tiempo de cómputo en el DGE (a partir del análisis de su costo computacional antes descrito). Además, el Lwf-CNN-lgR realiza el aprendizaje de reglas lingüísticas con un costo de  $O(n)$ . Concluyendo, el CRF es más eficiente que el Lwf-CNN-lgR por un menor consumo de memoria y tiempo durante su entrenamiento.

Esta desventaja en la eficiencia del Lwf-CNN-lgR es equilibrada por las siguientes ventajas:

- A diferencia del CRF, no es necesario el uso de ingeniería de características para el entrenamiento del conjunto de datos. Este tipo de ingeniería consume un gran tiempo de análisis y consulta de los datos a entrenar por parte de especialistas en PLN.
- El uso de GPU o un clúster de cálculo de alto desempeño (High-Performance Computing; *HPC*) para el entrenamiento de los modelos reduce el costo en tiempo y recursos

## **2.2 Análisis experimental del método para la extracción de aspectos**

---

de cómputo, de días a horas [139].

- El costo del entrenamiento de este modelo se compensa por los resultados de efectividad mostrados.
- Al obtener el modelo entrenado, su empleo en sistemas de recuperación de información o PLN se reduce a  $O(1)$  (e.d., el tiempo al predecir si el token de una palabra es un aspecto en la oración en la que aparece).

## **2.2. Análisis experimental del método para la extracción de aspectos**

Para evaluar el modelo propuesto, se seleccionaron siete conjuntos de datos de [62]; con opiniones en dominios como: reproductores de mp3, dvd, cámaras digitales y teléfonos inteligentes. Además, dos conjuntos de datos de [49] (e.d., uno sobre opiniones acerca de restaurantes y otro sobre laptops), y otro sobre opiniones de hoteles del sitio TripAdvisor<sup>7</sup>.

Las reglas lingüísticas fueron aplicadas a un conjunto de datos no etiquetado, usado en [140] para aumentar el conjunto de entrenamiento, que este conjunto proporciona 1000 opiniones obtenidas desde Amazon sobre 50 tipos de dispositivos electrónicos, tales como teclados, tablets, etc. El resultado de aplicar el algoritmo no supervisado de reglas lingüísticas, es la obtención de un conjunto de datos con las etiquetas que se infieren y el aprendizaje de nuevos patrones a partir de los datos del conjunto de entrenamiento.

Se diseñaron experimentos para comparar el desempeño de la propuesta con respecto a otros modelos del estado del arte:

- **Lifelong CRF:** La estrategia presentada en [62] que usa un modelo (*Conditional Random Field; CRF*) con una estrategia de AC.
- **CRF:** Un modelo CRF evaluado en una estrategia de aprendizaje multitarea que, durante el entrenamiento, tiene en cuenta todos los dominios [141].
- **Lwf-CNN:** La propuesta presentada en este trabajo con el modelo CNN y la estrategia de Aprendizaje Profundo sin las reglas lingüísticas.

---

<sup>7</sup><http://times.cs.uiuc.edu/wang296/Data>

## **2.2 Análisis experimental del método para la extracción de aspectos**

---

- **Lwf-CNN-lgR:** La propuesta presentada en este trabajo con el modelo CNN y la estrategia de AP con las reglas lingüísticas.

Los parámetros del modelo usaron, en su inicialización, una distribución uniforme U(-0.05, 0.05). La selección de este tipo de distribución está relacionada con la forma empleada por varios investigadores sobre el tema, como lo muestra el análisis del estado del arte empleado en esta investigación [8, 19, 43]. La tasa de aprendizaje fue 0.01 (respetando el valor usado en la propuesta de [40], teniendo en cuenta, además, que otros valores experimentales no mostraron mejor desempeño) con la función de optimización Adam para el cálculo del descenso del gradiente.

El valor utilizado para los ciclos de entrenamiento fue 100 y un lote con un valor de 12. El valor de  $\lambda_{prev}$  es igual a 0,0056 para controlar la influencia de dominios anteriores durante el aprendizaje del dominio actual. Se tomaron los mismos valores propuestos por los autores en [40].

Como medida de evaluación se seleccionó el F1-macro (F1), debido a su amplio uso en ABSA [8, 18] y que también fue empleada en la propuesta de [62]. Se realizaron dos tipos de pruebas o experimentos:

- *Pruebas de dominio cruzado:* Combinan seis conjuntos de datos etiquetados para el entrenamiento. Se realizan las pruebas en un séptimo dominio (no usado en el entrenamiento).
- *Pruebas en el dominio:* Se entrena y prueban los seis dominios, excluyendo un séptimo.

Aunque el enfoque de esta investigación está orientado al dominio cruzado (e.d., AC), se realizaron *las pruebas en el dominio* (e.d., como se enuncia anteriormente) para comparar con los resultados de [62].

En la figura 2.3 se muestran los resultados de la medida F1 correspondientes a *las pruebas con dominio cruzado*. Cada dominio indicado en el eje x fue el excluido durante el entrenamiento y usado como dominio de prueba. Los resultados muestran que el modelo Lwf-CNN-lgR obtiene los mejores resultados.

Esto es debido al uso del modelo CNN y el AC como principales características del modelo propuesto Lwf-CNN-lgR. Para las pruebas de dominio cruzado se demuestra el aporte del modelo para el AC porque obtiene mejores resultados al ser evaluado para instancias que no

## **2.2 Análisis experimental del método para la extracción de aspectos**

fueron vistas en el entrenamiento, pero pertenecen a un dominio muy cercano a los que fue entrenado el modelo. Confirmando que se aprendieron patrones comunes entre los objetos pertenecientes a cada conjunto de datos.

Por otro lado, en la figura 2.3 se muestran los resultados para la medida F1 correspondientes a las *pruebas en el dominio*. En este caso, los seis dominios restantes al indicado en el eje X se usaron tanto en el entrenamiento como en la prueba. Los resultados corroboran la validez del modelo propuesto Lwf-CNN-lgR, como el de mejor desempeño. Esto es debido a las características anteriormente enunciadas.

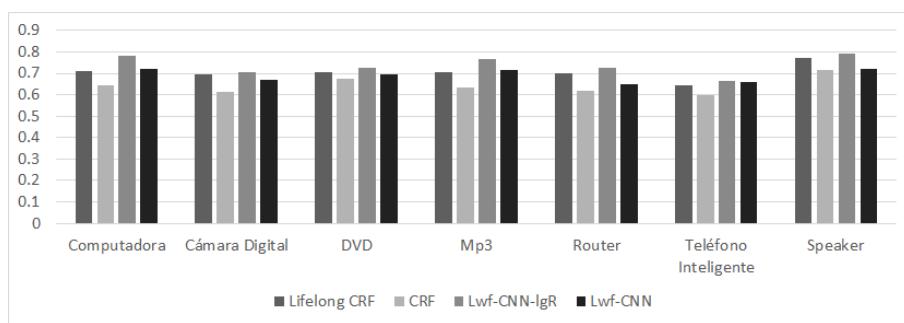


Figura 2.3: Resultados de la medida F1 con dominio cruzado.

El Lwf-CNN-lgR se beneficia de la coincidencia de palabras a través de dominios similares, así como del conjunto de datos no etiquetado, sobre dispositivos electrónicos.

El modelo propuesto es el mejor validar el uso de reglas lingüísticas, como forma de aumentar los conjuntos de datos y el aprendizaje de nuevo conocimiento de forma no supervisada; a pesar de su aporte no es significativo a la eficacia; pero su inclusión, como parte del proceso de aprendizaje, no degrada el desempeño de la propuesta presentada porque aportan a la efectividad del modelo en varios casos.

Con estos resultados se cumple el objetivo de comparar el uso de reglas lingüísticas para la extracción de aspectos. Este resultado debe ser investigado con mayor profundidad en estudios posteriores.

Para un análisis más exhaustivo de los resultados se realizó un segundo bloque de experimentos: a los conjuntos de datos asociados a dominios de efectos electrodomésticos, se añadieron

## 2.2 Análisis experimental del método para la extracción de aspectos

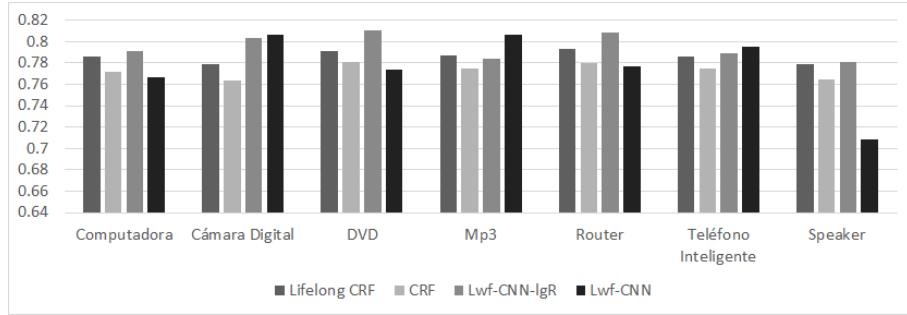


Figura 2.4: Resultados de la medida F1 en las pruebas en el dominio.

otros dos, formados por opiniones sobre restaurantes y hoteles, para analizar el comportamiento del modelo propuesto cuando los dominios son más diversos.

Los modelos Lwf-CNN y Lwf-CNN-lgR fueron evaluados en un esquema de dominio cruzado, usando los siete dominios (e.d., cuatro sobre productos electrodomésticos, uno sobre restaurantes, otros sobre laptops y otros sobre hoteles).

Los valores de F1-macro están por debajo del promedio de los resultados logrados en las experimentaciones anteriores, lo cual evidencia que la propuesta es sensible aún a la diversidad de dominios (la aparición del olvido catastrófico), como se muestra en la Tabla 2.1. Los bajos valores de calidad son causados por la existencia de nuevos aspectos en los conjuntos de datos de opiniones sobre restaurantes y hoteles, que no están asociados a los conjuntos de datos de opiniones sobre dispositivos electrónicos.

Tabla 2.1: Resultados en dominio cruzado para opiniones de restaurantes y hoteles.

Entrenamiento	Pruebas	Lwf-CNN-IgR			Lwf-CNN		
		P	R	F1	P	R	F1
-Restaurante	Restaurante	0.74	0.53	0.62	0.69	0.46	0.56
-Hotel	Hotel	0.78	0.59	0.67	0.62	0.48	0.54

Como tercer bloque de experimentos, dos dominios semánticamente cercanos (reproductor de DVD y Mp3) y no tan cercanos (Laptops y reproductor de DVD) fueron seleccionados para determinar si se produce el olvido catastrófico durante el entrenamiento de un nuevo dominio.

## **2.2 Análisis experimental del método para la extracción de aspectos**

Para ejecutar este experimento, el modelo fue primeramente entrenado con un dominio y luego con el otro. Después se ejecutó la prueba para el conjunto de pruebas del último dominio. Se demostró que es posible mantener resultados de desempeño aceptables, como se muestra en la figura 2.5 (se muestran los resultados de las medidas de desempeño Exactitud (P), Exhaustividad (R), además del F1-macro). Sin embargo, los mejores resultados se obtienen cuando se realizan las pruebas de los dominios semánticamente cercanos (e.d., aparecen términos de opinión y aspectos cercanos).

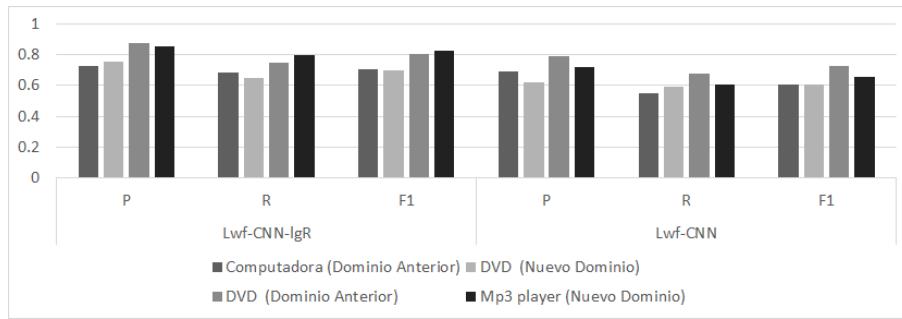


Figura 2.5: Resultados del experimento sobre la reducción del olvido catastrófico.

Tomando como base los resultados anteriores, podemos afirmar que la nueva propuesta mejora a la presentada en [62]. El uso de las reglas lingüísticas permite incrementar el conjunto de datos de entrenamiento, logrando mejorar los resultados de métodos de AP, cuando se tienen pocos datos para el entrenamiento. El modelo CNN propuesto en la investigación mejora, además, los resultados del modelo CRF.

Una de las ventajas del marco de trabajo propuesto es que no necesita de la ingeniería de características en el modelo de AP y AC. Este marco resalta la importancia de combinar un modelo de AP en un modo supervisado con patrones o reglas lingüísticas en la tarea de extracción del ABSA.

El uso práctico de este modelo se orienta a ser parte del pre-procesamiento de oraciones en idioma inglés (en esta investigación se evaluó, para estudios futuros, la factibilidad de construir un modelo computacional a partir del propuesto para el idioma español, ver Epígrafe 3.3.3).

Por ejemplo, en el texto de una opinión se obtienen las oraciones presentes; por cada una de ellas se reconocen las entidades nombradas [46]; de las palabras presentes en cada oración y

## **2.3 Propuesta de modelo computacional para la clasificación de aspectos**

mediante el modelo propuesto, se extraen las que son aspectos y no son entidades nombradas.

Para determinar la posible polaridad de los aspectos identificados por el modelo, es necesario crear un clasificador de aspectos. En las próximas secciones se propone un modelo con este objetivo, centrado en reducir el olvido catastrófico y mejorar los resultados de otras propuestas del estado del arte.

### **2.3. Propuesta de modelo computacional para la clasificación de aspectos**

Al realizar el procesamiento de la información textual asociada a una opinión, en un sistema de recuperación de información para el análisis de sentimientos, la última fase es la clasificación de la opinión empleando un modelo computacional (e.d., a través del uso de heurísticas de PLN, un modelo supervisado o no supervisado, o un modelo que combine estas estrategias).

En el caso de ABSA, es necesario la clasificación de la polaridad de las palabras identificadas como aspectos presentes en el texto de las oraciones de la opinión.

En esta sección se propone un nuevo método para la clasificación de aspectos, empleando el AC en la subtarea ABSA con las siguientes características:

- Adapta, para la clasificación en ABSA en un entorno DIL, el modelo AR1 [39] de regularización en el AC creado para la clasificación de imágenes.
- La capa de entrada al pre-modelo BERT es la representación de los *tokens* de la oración y los aspectos presentes en esta.
- Emplea la salida de BERT como entrada del modelo de AC.
- Realiza el ajuste de los pesos de la red neuronal en BERT durante el proceso de aprendizaje.
- La preservación de pesos relevantes en la red neuronal se realiza durante el cálculo del descenso del gradiente.
- El uso de la salida del modelo pre-entrenado BERT, mejora los resultados con respecto a otras formas de representación como Word Embeddings (p.ej., Glove [27]).

## **2.3 Propuesta de modelo computacional para la clasificación de aspectos**

---

Este modelo está compuesto por cuatro etapas. La etapa final ofrece un modelo listo para ser utilizado en un sistema de recuperación de información:

**Etapa 1: Representación textual:** Recibe la opinión textual original y retorna el vector de *tokens* para cada oración y las posibles palabras candidatas a ser clasificadas como aspectos (sustantivos, adjetivos y frases sustantivas) al aplicar un divisor de oraciones y un etiquetador morfológico con la herramienta spaCy. El modelo computacional, propuesto en la sección anterior, es usado para identificar los posibles aspectos.

**Etapa 2: Entrenamiento del modelo:** Aprende la información a incluir en la base de conocimientos (BC) dependiendo de BERT y el proceso de entrenamiento del modelo de AC para cada dominio. La etapa se divide en los siguientes pasos:

1. Entrenamiento de la red neuronal BERT para cada dominio, donde la última capa (salida) tiene tres neuronas (Positiva: P, Negativa: N y Neutral: Neu).
2. Entrenamiento del modelo de AC.

La salida de esta etapa está formada por los nuevos parámetros obtenidos en el entrenamiento.

**Etapa 3: Actualizar la base de conocimientos:** Aquí el olvido catastrófico es evitado a través del análisis de los resultados del proceso de entrenamiento. El error/pérdida obtenido es empleado en la optimización de los pesos por la estrategia de regularización. La salida es la actualización de la BC con los nuevos pesos obtenidos por el AC y la red neuronal de BERT.

**Etapa 4: Creación del clasificador de aspectos:** Se hace posible el uso del modelo de AC para resolver el ABSA en múltiples dominios. La configuración final es obtenida de los parámetros almacenados en la BC.

### **2.3.1. BERT como capa de vectores embebidos**

Las capas de vectores embebidos tradicionales Word2Vec o GloVe ofrecen una representación independiente del contexto para cada *token* [142]. Por el contrario, en BERT, la representación de estos está relacionada con los datos obtenidos en la oración empleada como entrada [31]. Esto permite tener más información acerca de las palabras del contexto cuando se entrena el

## 2.3 Propuesta de modelo computacional para la clasificación de aspectos

modelo.

En la primera etapa, las palabras de cada oración del conjunto de entrenamiento son usadas como la entrada del modelo pre-entrenado BERT. Como resultado, se obtiene un vector de pesos, que es la primera capa del modelo base de AP. En la propuesta se usa un modelo BERT pre-entrenado no sensitivo a casos<sup>8</sup>.

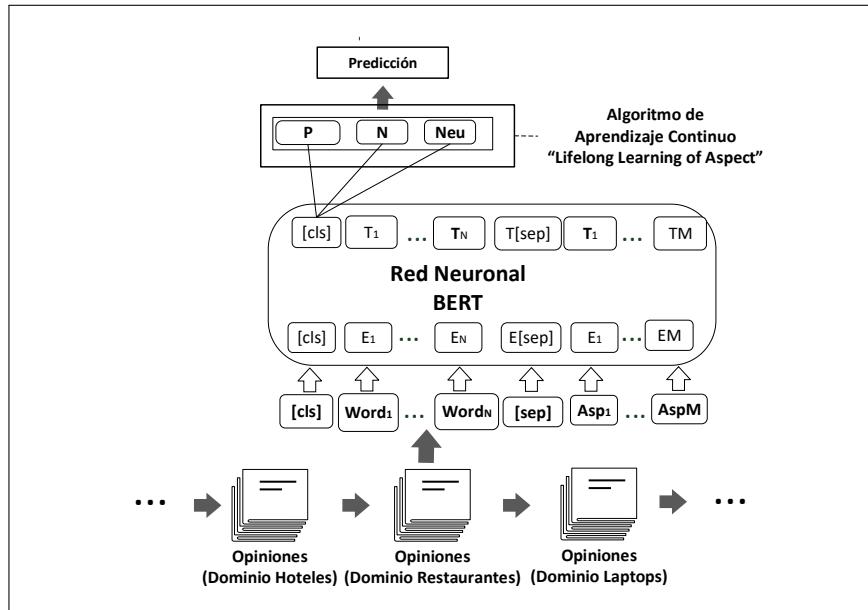


Figura 2.6: Representación de la información de entrada y salida durante el proceso de aprendizaje (P: positivo, N: negativo, Neu: neutral).

### 2.3.2. Diseño del modelo base para la clasificación de aspectos

El modelo propuesto aparece en la figura 2.6, donde se muestra el modelo base nombrado en Red Neuronal BERT y cómo este entrega su salida a partir de su neurona de clasificación (ver características del modelo BERT, propuestas en [82]) a una capa de tres neuronas completamente conectadas que constituye el modelo de AC.

BERT Special (*BSp*) [32] fue seleccionado como modelo base, y parte de la estrategia de AC presentada en este trabajo. Esto es debido a las ventajas en la representación de la información y su modelo de AP nombrado Transformer [82]. La arquitectura del modelo propuesta en esta

<sup>8</sup>no sensitivo a casos: El texto es llevado a minúsculas y se eliminan símbolos como acentos o diéresis.

## **2.3 Propuesta de modelo computacional para la clasificación de aspectos**

---

investigación permite su sustitución por cualquier otro modelo base de AP. Para demostrarlo, se realizó un conjunto de experimentos, que son descritos más adelante y cuyos resultados avalan esta posibilidad.

El modelo *BSp* construye una secuencia de entrada como la propuesta en [143]:

$$\langle \text{CLS} \rangle + \text{palabras de la oración} + \langle \text{SEP} \rangle + [\text{asp}] + \langle \text{SEP} \rangle \quad (2.12)$$

El primer *token* de cada secuencia de entrada es el vector especial de clasificación ( $\langle \text{CLS} \rangle$ ) y son separadas las palabras del contexto asociadas a las *palabras de la oración* y las *palabras clasificadas como aspectos* [asp] con un *token* especial ( $\langle \text{SEP} \rangle$ ).

El uso de BERT como modelo base se justifica por los notables resultados que ha obtenido en varias tareas del PLN y en particular el ABSA, superando los obtenidos con otras formas de representación, como Word2Vec o GloVe [1]. La salida de la neurona con el *token*  $\langle \text{CLS} \rangle$  es la entrada de la última capa, que es clave para obtener el valor de clasificación en el proceso del aprendizaje de la segunda etapa de la construcción del modelo.

El *BSp* fue seleccionado porque combina un MA con una red neuronal como la de BERT, además en el vector de salida se asocian el valor de representación y la posición del *token* en la oración. Esto permite un aprendizaje más cercano al contexto donde aparece el aspecto en la oración [82].

La arquitectura simple del modelo base fue validada en los experimentos contra otros modelos más complejos como el *Attentional Encoder Network with BERT* (AE), el que obtiene un 0.73 de F1-macro para un conjunto de datos con opiniones sobre restaurantes [32].

### **2.3.3. Modelo para la clasificación de aspectos**

La propuesta presentada es nombrada *Lifelong Learning of Aspects* (LLA), está inspirada en la propuesta *Architectural and Regularization 1* (AR1) [39] y la nombrada *Synaptic Intelligence* (SI) [129], debido a sus buenos resultados comparados con otros modelos [38] del estado del arte. Ambos modelos fueron aplicados en la clasificación de imágenes. Uno de los principales aportes de esta investigación fue adaptar los principales conceptos de AR1 y SI al PLN. Las adaptaciones realizadas fueron:

## **2.3 Propuesta de modelo computacional para la clasificación de aspectos**

---

- El uso de un modelo base orientado al PLN como BERT a diferencia de una arquitectura CNN o un modelo pre-entrenado como Resnet para el procesamiento de imágenes.
- El modelo AR1 fue entrenado y evaluado en un contexto TIL (e.d., para cada nueva tarea o dominio durante el AC aparecen nuevas clases). En contraste con el modelo AR1, la propuesta LLA fue adaptada al etiquetado de palabras en oraciones, y se crearon tres clases de clasificación (positivo, negativo y neutral) para diferentes conjuntos de datos o dominios.

Aunque AR1 mejora al SI en la clasificación de imágenes [39], para la definición del modelo LLA los principales conceptos fueron tomados de SI. Esta decisión se basa en la necesidad de evitar grandes cambios en el peso de las neuronas de la arquitectura de red durante el proceso de aprendizaje continuo de nuevos dominios.

El algoritmo 2 representa el proceso de entrenamiento del modelo LLA, el cual tiene como entrada, cada una de las oraciones del conjunto de datos, la palabra que representa al aspecto (presente en esta oración) y su valor de polaridad.

Los conjuntos de tokens representan a las palabras de estas oraciones en un vector de valores reales por el modelo BERT (modelo base del LLA). La salida es un vector de valores reales donde las dimensiones representan a cada uno de los tokens de entrada, más un token asociado a la clasificación. Este valor es tomado como entrada a una capa de red neuronal completamente conectada, que es la que utiliza en su entrenamiento el modelo LLA.

En el entrenamiento secuencial de dominios, los pesos de la última capa que pertenece al modelo LLA, relacionada con la clasificación, solo son inicializados a 0 para el primer dominio. Durante el entrenamiento se ajustan los pesos del modelo base BERT y del modelo de AC nombrado LLA, esto se realiza con el objetivo de aprender a reconocer mejor las opiniones asociadas a los dominios y mejorar la efectividad del modelo de lenguaje BERT para la tarea ABSA.

Al realizar la clasificación de un aspecto según su polaridad , la función de costo realiza una compensación entre el nuevo valor del peso de la neurona y el obtenido durante el entrenamiento del ejemplo anterior (para las oraciones de un mismo dominio o entre dominios). Esto evita que se realicen grandes modificaciones y se pierda el conocimiento aprendido anteriormente.

El principal objetivo en LLA es obtener un conjunto de pesos en la capa de salida  $\vec{c}w$ , como

## 2.3 Propuesta de modelo computacional para la clasificación de aspectos

---

### Algorithm 2 Lifelong Learning of Aspects (LLA)

---

**Entrada :**

$\vec{c}w = <0, 0, 0>$	▷ Vector de pesos usados para la inferencia.
$\bar{M} = 0$	▷ Conjunto de pesos del modelo base de aprendizaje profundo.
$M = 0$	▷ Conjunto de pesos del modelo base compartidos durante el entrenamiento.
$\hat{F} = 0$	▷ Matriz de pesos relevantes usada por el algoritmo Synaptic Intelligence.
$Text$	▷ Conjunto de oraciones (e.d., conjunto de datos de cada dominio).
<b>Salida :</b> $\vec{c}w$	▷ Vector de pesos que serán empleados en la inferencia
$\bar{M}$	▷ Conjunto de pesos del modelo base (autoajuste de BERT)

- 1: Dado  $Text$  extraer cada oración  $x$  y sus aspectos candidatos  $y$ , dividirlos en subconjuntos (lotes  $B$ ).
  - 2: **loop** ▷ Para cada subconjunto tomar los pares  $(x, y)$
  - 3: Entrenar el modelo base con  $(x, y)$  para las clases: positivo negativo, neutral
  - 4: Aprender  $\bar{M}$  y  $\vec{c}w$  con el algoritmo de Regularización  $SI$  y los valores en  $\hat{F}$  y  $M$
  - 5: Actualizar los pesos  $\vec{c}w$  y los de  $M$  con  $M = \bar{M}$
  - 6: Actualizar  $\hat{F}$  según a los valores calculados en el lote actual
  - 7: Evaluar el modelo usando  $\bar{M}$  y  $\vec{c}w$
  - 8: **end loop**
- 

se muestra en el algoritmo 2. Una de las principales adaptaciones, inspirado en AR1, en el modelo LLA es que el  $\vec{c}w$  es inicializado a cero (como entrada) y no se emplea la inicialización aleatoria como en otros modelos AC.

La inicialización 0 en la capa de salida difiere de la típica inicialización Gaussiana o de Xavier, empleada en el entrenamiento de modelos de AP [19]. Los pesos de la red neuronal no deben ser inicializados a 0, porque pueden causar que la activación de las neuronas intermedias sea 0 y hacer nulos los efectos de la “propagación hacia atrás” (Backpropagation). Esto se cumple para valores de los pesos en niveles intermedios, pero no en el caso de la capa de salida (Ver demostración en el Anexo E del artículo propuesto por Maltoni *et al.* [144]). En el caso del modelo LLA, la capa de salida coincide con la de clasificación, que no es una capa inicial o intermedia, es en esta donde se realiza la inicialización a 0.

Los parámetros de cada capa de salida de los dominios previos son almacenados en  $\vec{c}w$ . El modelo base de AP es entrenado usando cada lote  $B$  de oraciones en el conjunto de datos, como se muestra en la línea 3.

El modelo de AC propuesto es descrito en las líneas 4 y 5. En estas, el vector  $\vec{c}w$  es la BC

## **2.3 Propuesta de modelo computacional para la clasificación de aspectos**

---

en el nuevo dominio de clasificación y el uso de la matriz de pesos importantes  $\hat{F}$  y el vector  $\vec{c}\vec{w}$  son la principal estrategia de regularización para reducir el olvido catastrófico, que ocurre porque al actualizar se establece un balance entre el proceso de aprendizaje en cada dominio y la evaluación del resultado de clasificación de las instancias del dominio actual.

En la línea 6 se actualizan los parámetros de la red neuronal empleando la regularización y el DGE.

A partir de la ecuación 2.3, que define la función de costo de un modelo de AP, enunciada en el epígrafe 2.1.1, se establece como función de costo para el modelo *LLA*:

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha G(\theta) \quad (2.13)$$

Donde  $G(\theta)$  es igual a  $\Omega(\theta)$ , en la ecuación 2.3, y define:

$$G(\theta) = \sum_t \Phi_t \left( \beta'_t - \beta_t \right)^2 \quad (2.14)$$

Para la estimación del error de la predicción  $J(\theta; X, y)$  se procede de igual forma que en la ecuación 2.7 para un lote o subconjunto de ejemplos del conjunto de datos o dominio de entrenamiento y se promedia el resultado obtenido.

Para el término de regularización  $G(\theta)$ ,  $\beta_t$  es el  $t$ -ésimo parámetro en la arquitectura de red neuronal (la capa de clasificación y en el modelo BERT) de LLA,  $\alpha$  es un parámetro de penalización que compensa los resultados obtenidos en las tareas antiguas (dominios) con la nueva o actual,  $\Phi_t$  representa la fuerza de regularización para los  $t$  parámetros del modelo completo.

La expresión  $(\beta'_k - \beta_k)$  es la variación o diferencia entre  $\beta'_t$  (el peso del  $t$ -ésimo parámetro en el entrenamiento del dominio anterior) y  $\beta_t$  en el del dominio actual.

El modelo LLA, a diferencia del Lwf-CNN-lgR, no necesita preservar las capas de clasificación de los dominios anteriores, porque solo usa como referencia los valores de los parámetros del dominio anterior. Esto es una mejora en eficiencia porque reduce el tiempo de cómputo y la memoria necesaria durante el entrenamiento del modelo.

El seudo-código asociado a la ecuación 2.13 se puede representar como:

## 2.3 Propuesta de modelo computacional para la clasificación de aspectos

---

### Algorithm 3

---

```

1: modelRegultarization = 0
2: for  $t_i \leftarrow \text{range}(1, t)$  do                                ▷ Para todos los parámetros del modelo
3:    $\text{modelRegultarization} += \text{regularizationStrong}(t_i) * (t_i^{old} - t_i)$ 
4: end for
5:  $loss_{ce} = ce(outputs, targets)$                       ▷ Cálculo de la entropía cruzada
6: return  $loss_{ce} + lamb * \text{modelRegultarization}$ 

```

---

Donde  $t_i^{old}$  representa el  $t$ -ésimo parámetro en el entrenamiento del dominio anterior, que debe ser preservado hasta que se termine el aprendizaje del dominio actual. En la llamada a la función  $\text{regularizationStrong}(t_i)$  se realizan todos los cálculos asociados a  $\Phi_t$  que se describen en los párrafos siguientes.

En esta propuesta, el uso del algoritmo Synaptic Intelligence (SI) es la principal estrategia para preservar los pesos aprendidos en las tareas anteriores. Este algoritmo calcula la importancia de los pesos durante la actualización de pesos en el DGE. La ecuación 2.15 actualiza los pesos  $\Delta\mathcal{L}_t$ , donde  $\Delta\theta_t = \theta'_t - \theta_t$  es la variación de los pesos actuales e iniciales, y  $\frac{\partial\mathcal{L}}{\partial\theta_t}$  es el valor del gradiente.

$$\Delta\mathcal{L}_t = \Delta\theta_t \cdot \frac{\partial\mathcal{L}}{\partial\theta_t} \quad (2.15)$$

Los cambios asociados al parámetro  $\theta_k$  se obtienen al ejecutar la suma  $\Sigma\Delta\mathcal{L}_t$  sobre la trayectoria de pesos (p.ej., la secuencia de pasos de actualización durante el entrenamiento del lote).

La fuerza de regularización (denotada como  $\Phi_t$ , representada en la ecuación 2.13, se obtiene como se muestra en la ecuación 2.16, donde  $T_t$  es el movimiento total del  $\theta_t$  durante el entrenamiento de un lote (p.ej., la diferencia entre un valor final e inicial) y  $\xi$  es una constante para evitar la división por cero (ver [129] para más detalles)).

$$\Phi_t = \frac{\Sigma\Delta\mathcal{L}_t}{T_t^2 + \xi} \quad (2.16)$$

Toda la información necesaria para calcular  $\Phi_t$  está disponible durante el cálculo del DGE

## **2.4 Medidas para evaluar la calidad del aprendizaje de los modelos**

---

y no se requieren otros cálculos como en otros modelos más costosos (p.ej., EWC). En la ecuación 2.17,  $\Phi$  se le asigna cero antes del primer lote o subconjunto de ejemplos del conjunto de entrenamiento, y luego se modifica como la suma de los parámetros  $\theta$  del modelo en un lote específico.

$$\Phi = \begin{cases} 0, & \text{Antes de todos los lotes} \\ \Phi(index - 1) + \sum_i^t \theta_i \Phi_t, & \text{en el lote con el número } index \end{cases} \quad (2.17)$$

Luego se calcula  $\hat{\Phi}$  mediante la función  $clip(\Phi, max)$ , devolviendo los valores de la matriz que exceden la constante  $clip$ .

El proceso de actualización de los pesos de los parámetros del modelo constituye la segunda etapa del proceso de construcción del modelo de clasificación de aspectos. En la etapa siguiente se actualiza la BC. El proceso de entrenamiento finaliza (cuarta etapa) al concluir el proceso de aprendizaje del modelo con el último conjunto de datos (dominio). Al concluir estas etapas, es necesario determinar la efectividad o calidad del modelo para la clasificación de aspectos.

## **2.4. Medidas para evaluar la calidad del aprendizaje de los modelos**

La medida de exactitud (accuracy) es muy usada para evaluar los modelos en ABSA [8, 18]. Otras medidas que permiten determinar la efectividad de un modelo son exhaustividad (recall) y F1-macro. Estas permiten mostrar los resultados de las clases que tienen menos instancias en el conjunto de datos [5, 16]. Teniendo en cuenta que los conjuntos de datos seleccionados para realizar la evaluación no están balanceados (Ver la Tabla 2.3), para estimar el desempeño, durante la experimentación, se usó el F1-macro (el F1-score es promediado sobre todas las clases) junto con la exactitud.

Otra medida usada fue la Cohen-Kappa (Kappa) [145], que permite considerar la efectividad de un modelo entrenado en un conjunto de datos desbalanceado [51]. Kappa es calculada como se muestra en la ecuación 2.18, donde  $\rho_o$  es la probabilidad de que una etiqueta de clase sea

## 2.4 Medidas para evaluar la calidad del aprendizaje de los modelos

---

asignada a una instancia del conjunto de datos, y  $\rho_e$  es la etiqueta de clase real que tiene la instancia:

$$\mathcal{K} = (\rho_o - \rho_e) / (1 - \rho_e) \quad (2.18)$$

Con Kappa se obtienen valores en el intervalo  $[-1, 1]$ , donde valores iguales o menores que cero significan que el modelo no tiene valores relevantes [51].

### 2.4.1. Medidas para estimar el olvido catastrófico en modelos de aprendizaje continuo

Para estimar la efectividad del AC en los dominios anteriores, algunos autores han empleado la medida de Transferencia hacia Atrás (*Backward Transfer*; BWT) [16]. Valores de  $BWT < 0$  indican la presencia de olvido catastrófico, y  $BWT \geq 0$  indican que el modelo ha podido mejorar el desempeño en las tareas anteriores, después de entrenar en la nueva tarea [39]. Sin embargo, es frecuente generar valores negativos de  $BWT$  por la naturaleza de un posible olvido en los modelos de AP (e.d., por el uso de redes neuronales computacionales como principal estructura en sus modelos).

$$BWT_r = \frac{\sum_{i=2}^n \sum_{j=1}^{i-1} (R_{i,j} - R_{j,j})}{\frac{n(n-1)}{2}} \quad (2.19)$$

Al analizar la ecuación para la estimación del  $BWT$ , podemos determinar que siempre se resta la evaluación del modelo (exactitud o F1-macro) de los conjuntos de pruebas de las tareas anteriores ( $i$ ) con respecto al modelo entrenado en el conjunto de datos actual ( $j$ ) a la eficacia de la evaluación en el dominio actual (e.d., representado por  $(j, j)$ ). Una conclusión *a priori* es que los resultados del modelo, al ser entrenado y evaluado en el conjunto de datos actual  $(j, j)$ , obtendrá valores altos de calidad, porque la red neuronal está aprendiendo los pesos con los datos actuales y el proceso de aprendizaje tiene poca influencia de los valores de los pesos obtenidos en tareas anteriores.

La medida  $BWT$  ha sido empleada para la evaluación de modelos en el área del procesamiento de imágenes y donde la tarea del AC consiste en estimar diferentes clases a partir del mismo conjunto de datos (p.ej., estimar para un mismo conjunto de datos primero imágenes de perros y gatos; después si las imágenes corresponden a uno o varios animales, etc.).

## **2.4 Medidas para evaluar la calidad del aprendizaje de los modelos**

---

Las tareas de procesamiento de imágenes y del PLN se diferencian en varias características; una de las más notables es la naturaleza secuencial del PLN, por lo que se deben seleccionar medidas para el olvido catastrófico que la tengan en cuenta.

En [146] se propone la medida representada por la ecuación 2.20 y empleada en trabajos orientados al análisis de sentimientos (p.ej., ABSA) como en [2].

$$F_T = \frac{1}{T-1} \sum_{i=1}^{T-1} (f_i^T) \quad (2.20)$$

En esta ecuación  $f_i^T$  está definida por:

$$f_i^T = \max_{l \in 1, \dots, T-1} a_{l,j} - a_{T,j}, \forall j < T \quad (2.21)$$

En la ecuación 2.21,  $a$  representa el valor de la medida de calidad al evaluar el modelo en el conjunto de prueba de uno de los dominios aprendidos,  $f_i^T \in [-1, 1]$  y se define  $j < T$ , para cuantificar el olvido catastrófico en tareas anteriores.

Esta medida se define a partir de la crítica hecha a la medida presentada por [120] y que es la base del *BWT* propuesto en [39]. Nótese que en la ecuación 2.20 se mejora las carencias del *BWT*, debido a que se evalúa el valor del modelo con los conjuntos de pruebas de los dominios anteriores. Esto permite analizar si existe olvido catastrófico sin considerar los resultados del conjunto de prueba del dominio recientemente aprendido.

La medida usada en [131] se ajusta a los objetivos de esta investigación porque es empleada en problemas de análisis de sentimientos en oraciones. Esta medida llamada en [131] *estimación del aprendizaje hacia atrás* (sin olvidar el aprendizaje en tareas anteriores), consiste en promediar el resultado del modelo (entrenado en la tarea actual) en los conjuntos de prueba de las tareas anteriores a la última, entrenada por el modelo de AC.

Varias de estas medidas han sido empleadas en problemas de clasificación de imágenes y en diferentes contextos de AC (aprendizaje de nuevos dominios para el mismo conjunto de clases, aprendizaje de nuevos dominios y clases, etc.), ajustándose al propósito de cada tarea.

Es un reto de investigación lograr una medida estándar de olvido catastrófico [25, 33, 36]. A partir del análisis anterior, se seleccionó la medida *estimación del aprendizaje hacia atrás* para

## **2.5 Conjuntos de datos empleados para el aprendizaje de los modelos**

---

estimar el olvido catastrófico del modelo propuesto, teniendo en cuenta su coincidencia con los objetivos a evaluar. La efectividad de la clasificación se analiza a través de los resultados de las medidas Kappa y F1-macro, teniendo en cuenta el desbalance de los conjuntos de datos de entrenamiento (Ver ejemplos en la Tabla 2.3) y la exactitud.

### **2.5. Conjuntos de datos empleados para el aprendizaje de los modelos**

Para entrenar los modelos y evaluar su desempeño, los experimentos se realizaron empleando siete de los conjuntos de datos más usados en ABSA. Estos se describen en la Tabla 2.2 y fueron tomados de cuatro fuentes diferentes.

<b>Dominios</b>	<b>Oraciones</b>	<b>Aspectos</b>	<b>oraciones de entrenamiento</b>	<b>oraciones de prueba</b>
Cámaras digitales	597	237	477	120
Teléfonos inteligentes	546	302	436	110
Routers	701	307	877	176
Restaurantes	3841	4722	3041	800
Laptops	3845	2951	3045	800
Hoteles	4856	3810	3371	1485

Tabla 2.2: Descripción de los conjuntos de datos usados.

El conjunto de opiniones sobre laptops y restaurantes fue tomado de la tarea 4 de la competición internacional SemEval-2014 [49]; el conjunto de datos sobre dispositivos electrónicos fue tomado de [147, 148]; mientras que las opiniones sobre hoteles fueron tomadas de TripAdvisor [143]. Los conjuntos de entrenamiento y pruebas fueron definidos por sus autores.

## 2.5 Conjuntos de datos empleados para el aprendizaje de los modelos

Dominios	Positivo	Negativo	Neutral
Restaurantes	2892	1001	829
Laptops	1328	994	629
Hotels	2343	656	811

Tabla 2.3: Ejemplo del desbalance entre las clases en los conjuntos de datos.

### 2.5.1. Estudio de la cercanía semántica de los conjuntos de datos

Los conjuntos de datos (dominios), empleados en el aprendizaje del modelo y con los que se comparan otras propuestas del estado del arte (*State Of The Art*; SOTA), fueron creados por humanos. Pueden ser considerados agrupamientos porque relacionan opiniones sobre un mismo tema o dominio.

Para tener un conocimiento de la cercanía semántica entre estos dominios se calculó el centroide (el promedio de los vectores de salida de BERT de todas las oraciones en cada dominio) y se usó la similaridad coseno entre estos centroides [149] con el objetivo de estimar la cercanía entre ellos, como se muestra en la figura 2.7.

La similaridad (colores cercanos al amarillo o rojo) que se muestra entre dominios indica que los dominios de restaurantes y hoteles son cercanos. Sin embargo, otros como routers, laptops no son cercanos a restaurante.

Otra medida importante para evaluar la calidad de estos agrupamientos es el coeficiente de Silhouette [150]. Los valores de esta medida se encuentran en el intervalo  $[-1, 1]$ , y valores cercanos a cero indican solapamiento entre agrupamientos. El coeficiente de Silhouette (con respecto a la similaridad coseno) entre los conjuntos de datos fue de -0.017, lo que indica que existe solapamiento o términos en común entre dominios.

Este resultado permite suponer que existen términos o patrones en común entre dominios. Esto será demostrado por los resultados de la clasificación y reducción del olvido catastrófico en los experimentos del modelo de AC propuesto.

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

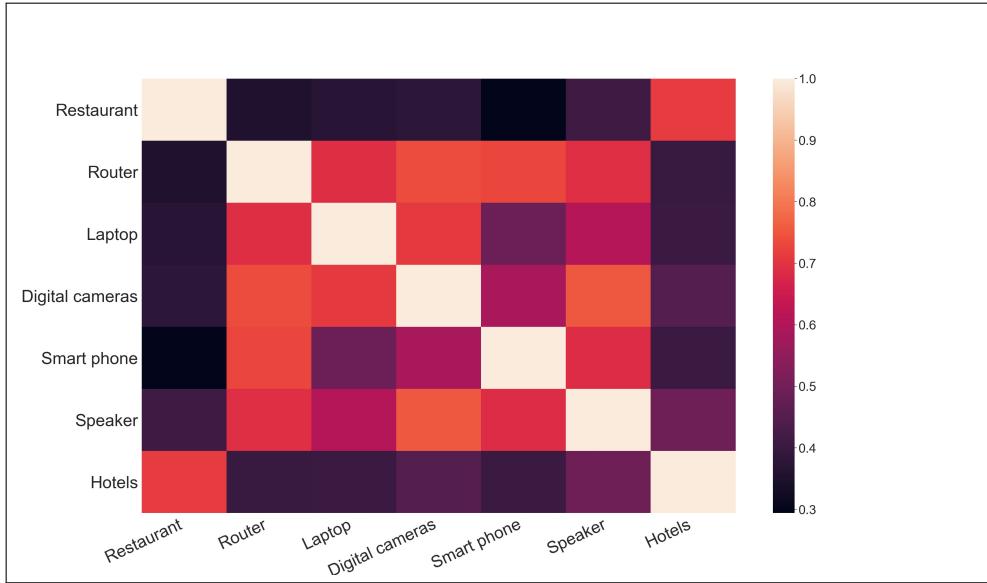


Figura 2.7: Similaridad coseno entre los centroides de cada dominio.

## 2.6. Evaluación de la propuesta para la clasificación de aspectos

Para evaluar el comportamiento del modelo LLA con respecto al estado del arte, se diseñó un conjunto de experimentos que permiten comparar su desempeño en la subtarea polaridad del sentimiento (*Sentiment Polarity*; SP) o clasificación de aspectos (CA) de ABSA con respecto a varias propuestas del estado del arte del AC [39].

### 2.6.1. Detalles de implementación de los experimentos

El vector pre-entrenado de Glove usado como Word Embeddings tiene una dimensión de 300<sup>9</sup>. El modelo pre-entrenado de BERT empleado para el entrenamiento fue BERT-Base Uncased<sup>10</sup> con 12-capas, 768-ocultas, 12-cabezas y 110 millones de parámetros.

Los pesos de los modelos LLM, EWC fueron inicializados con el método Glorot<sup>11</sup>, el coeficiente de la regularización  $L_2$  es  $10^{-5}$  y el valor de la regularización por omisión es 0.1 [32].

<sup>9</sup>Valor definido por los creadores de la herramienta.

<sup>10</sup>Uncased: El texto se ha puesto en minúsculas antes del paso de tokenización de WordPiece.

<sup>11</sup>Sigue una distribución uniforme  $U(-1, 1)$  en el momento de asignar los valores a los pesos iniciales de la red.

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

Todas las combinaciones de modelos de AP y AC han sido entrenadas con un lote de tamaño 64 y 10 ciclos de entrenamiento, para todos los conjuntos de datos. La función de optimización usada fue Adam con una tasa de aprendizaje de 2e-5<sup>12</sup>. Para la implementación de los modelos se empleó inicialmente la biblioteca pytorch-transformers<sup>13</sup> en su versión 1.2.0 y luego migrada con el uso de transformers<sup>14</sup> en su versión 2.1.0.

Los procesos de entrenamiento y evaluación fueron realizados sobre un 2 x Intel Xeon L5520 con 64 Gigabyte RAM en un clúster de Cálculo de Alto Desempeño (High-Performance Computing; HPC) de la Universidad Central “Marta Abreu” de Las Villas, Cuba. Los experimentos para evaluar propuestas recientes del estado del arte de clasificadores en ABSA con respecto a la propuesta de esta investigación se realizaron en el HPC de la Universidad Jaume I<sup>15</sup> de Castellón (Valencia, España) en dos Unidades de Procesamiento Gráfico (GPU; siglas en inglés) de la marca NVIDIA, versión v100<sup>16</sup> con 32 Gigabyte RAM cada una. El código de los modelos se encuentra público<sup>17</sup>.

### **2.6.2. Descripción de las experimentaciones**

En el diseño experimental, cada modelo de AP para ABSA fue usado como modelo base de un modelo de AC (combinación entre modelos de aprendizaje profundo y continuo). Entre los modelos comparados, el Lifelong Learning Memory Network (LLM) propuesto en [96] incluye el mecanismo de AC, por lo que no es necesario combinarlo con otro de AC.

El objetivo global es verificar la efectividad de la nueva propuesta con respecto a SOTA. Para lograrlo se siguieron las siguientes estrategias:

1. Comparar el modelo LLA con los principales modelos del estado del arte.
2. Analizar si el orden en que los dominios son procesados afecta la calidad de los resultados, debido a que el modelo aprende de forma secuencial.

Para seguir las estrategias de experimentación propuestas, las pruebas se realizaron con cada

---

<sup>12</sup>Estos valores coinciden con los empleados con los modelos comparados para lograr iguales condiciones de experimentación.

<sup>13</sup><https://pypi.org/project/pytorch-transformers>.

<sup>14</sup><https://pypi.org/project/transformers/2.1.0>.

<sup>15</sup><https://www.uji.es/>

<sup>16</sup><http://www.hpca.uji.es/node/2>

<sup>17</sup><https://github.com/dionis/ABSA-DeepMultidomain/>

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

una de las posibles secuencias de conjuntos de datos o dominios teniendo en cuenta su orden de aparición, y evaluando los conjuntos de pruebas del dominio que se aprende durante la iteración de la secuencia; promediando el resultado final para cada una de las medidas de evaluación.

Varios conjuntos de datos (cámaras digitales, routers, speakers, laptops) están relacionados semánticamente con el dominio de dispositivos electrónicos. Por esta razón, durante la experimentación, estos conjuntos no fueron permutados en el orden en que son aprendidos (e.d., se toma como un único gran conjunto)<sup>18</sup>.

El tiempo de ejecución de los métodos fue medido durante cada prueba. El orden de entrenamiento de los conjuntos de datos de opiniones para cada una de las pruebas fue:

- **REH:** Restaurantes, cámaras digitales, routers, speakers, laptops, Hoteles.
- **HER:** Hoteles, cámaras digitales,rRouters, speakers, laptops,restaurantes .
- **EHR:** Cámaras digitales, routers, speakers, laptops, hoteles, Restaurantes .
- **ERH:** Cámaras digitales, routers, speakers, laptops, restaurante, Hoteles .
- **RHE:** Restaurantes, hoteles, cámaras digitales, routers, speakers, laptops.
- **HRE:** Hoteles, restaurantes, cámaras digitales, routers, speakers, Laptops.

Dos tipos de configuraciones se tuvieron en cuenta durante los experimentos. Inicialmente, se analizó el desempeño de los modelos sin ajustar los pesos de la arquitectura de BERT. Esta experimentación obtuvo resultados de exactitud cercanos a 0.65; por ser muy bajos con respectos al estado del arte [1, 8] fue rechazada.

En la configuración final, los pesos de la arquitectura BERT y los del modelo de AC fueron ajustados durante los pasos de propagación hacia atrás (*Backpropagation*). Varios autores [83] explotan esta posibilidad para el entrenamiento de BERT con buenos resultados en ABSA al ser evaluados los modelos en dominios específicos (p.ej., opiniones sobre restaurantes).

---

<sup>18</sup>No se creó otro gran conjunto con las oraciones de opinión de hoteles y restaurantes porque tendríamos dos grandes conjuntos (origen-destino) y se transformaría en un problema de aprendizaje por transferencia que no es el objetivo de esta investigación.

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

### **2.6.3. Modelos del estado del arte para la evaluación del aprendizaje profundo y continuo**

Como modelos de AP en el proceso experimental fueron seleccionados dos modelos de clasificación empleados en ABSA: Local Context Focus with BERT (LC) [151] y Attentional Encoder Network with BERT (AE) [32]. El primero alcanzó 0.82 de exactitud con un conjunto de datos de opiniones de laptops, y la segunda propuesta obtuvo un 0.83 de exactitud con un conjunto de datos de opiniones de restaurantes. Estos modelos fueron seleccionados por su arquitectura y los altos valores de exactitud con respecto a SOTA. Además, tienen como entrada el vector de salida de un modelo pre-entrenado BERT. En el caso de AE incluye una propuesta que tiene como entrada el vector embebido obtenido a partir de un modelo pre-entrenado de Glove. A continuación, se describe cada modelo:

- *Local Context Focus with BERT (LC)* [151]: Recibe como entrada las palabras correspondientes a la oración donde el aspecto aparece y un conjunto de palabras en la vecindad del aspecto. La arquitectura de red neuronal de peso dinámico de características del contexto es propuesta como capa superior de la salida del modelo BERT.
- *Attentional Encoder Network with BERT (AE)* [32]: Hace uso de una arquitectura de atención de varias entradas (*Multi-Head Attention*) para la salida del modelo BERT. En este caso son dos entradas: las palabras del contexto (una oración) y las palabras que constituyen el aspecto.
- *Attentional Encoder Network with LSTM (AT)* [22]: Hace uso de un MA y concatena la representación de aspectos y su contexto. AT establece los aspectos que son parte del cálculo de los pesos de atención. Este usa el vector embedido pre-entrenado de Glove como entrada y LSTM como modelo de AP.

Para evaluar el modelo propuesto fueron tomadas de la literatura tres estrategias: *Lifelong Learning Memory* (LLM), *Elastic Weight Consolidation* (EWC), y *Architectural and Regularization 1* (AR1).

Las tablas 2.5-2.6 muestran el desempeño de SOTA y el nuevo modelo propuesto en esta investigación. La combinación de los métodos de aprendizaje profundo y continuo tienen la siguiente notación:

**⟨Acrónimo del método de aprendizaje profundo⟩⟨Acrónimo del método de aprendizaje continuo⟩**

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

---

(p.ej.,  $AE_{LLA}$ ).

Acrónimos	Modelos combinados
$LLM$	<i>Lifelong Learning Memory (LLM)</i>
$AE_{EwC}$	<i>Attentional Encoder Network con BERT y EwC</i>
$AE_{LLA}$	<i>Attentional Encoder Network con BERT y el nuevo modelo LLA</i>
$AE_{AR1}$	<i>Attentional Encoder Network con BERT and AR1</i>
$BSp_{EwC}$	<i>BERT Special con EwC</i>
$BSp_{LLA}$	<b><i>BERT Special con el nuevo modelo LLA</i></b>
$AT_{LLA}$	<i>Attentional Encoder Network con LSTM y el nuevo modelo LLA</i>
$LC_{LLA}$	<i>Local Context Focus con BERT y el nuevo modelo LLA</i>

Tabla 2.4: Acrónimos y nombres de los modelos comparados durante la evaluación de la nueva propuesta.

### 2.6.4. Evaluación de la estrategia de comparación con otros modelos del estado del arte

Teniendo en cuenta que los conjuntos de datos seleccionados son desbalanceados (ver ejemplo en la Tabla 2.3), se usó el F1-macro de conjunto con la exactitud (Accr) y el Cohen-Kappa (Kappa).

Modelo	$AE_{LLA}$	$AT_{LLA}$	$LC_{LLA}$	$BSp_{LLA}$
Accr	0.69	0.64	0.79	<b>0.80</b>
F1	0.49	0.38	0.66	<b>0.73</b>
Kappa	0.40	0.12	0.59	<b>0.62</b>
<b>OvrcForgtt</b>	0.49	0.38	0.66	<b>0.73</b>

Tabla 2.5: Promedio de los resultados al emplear diferentes modelos base y la estrategia de AC propuesta en esta investigación.

La medida seleccionada para evaluar el olvido catastrófico fue la propuesta en [131], y fue seleccionada porque este trabajo está orientado al AC del análisis de sentimientos en oraciones; muy cercano al objetivo de la investigación. La medida es llamada **OvrcForgtt** en los resultados mostrados en las tablas (p.ej., Tabla 2.6)

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

El modelo  $BSp_{LLA}$  obtiene los mejores resultados contra otros modelos base y la misma estrategia de AC (LLA), como se muestra en la Tabla 2.5.

Modelo	$LLM$	$AE_{EwC}$	$AE_{AR1}$	$BSp_{EwC}$	$BSp_{LLA}$
Accr	0.39	0.68	0.57	0.75	<b>0.80</b>
F1	0.23	0.50	0.33	0.62	<b>0.73</b>
Kappa	0.03	0.42	0.16	0.53	<b>0.62</b>
<b>OvrcForgtt</b>	0.23	0.49	0.32	0.62	<b>0.73</b>

Tabla 2.6: Promedio de los resultados entre  $BSp_{LLA}$  y otras propuestas en SOTA.

Estos resultados permiten identificar el aporte significativo del  $BSp$  como modelo base, al modelo final y demuestra que el cambio de este componente del modelo general puede ser sustituido con otro tipo de propuesta.

La comparación del  $BSp_{LLA}$  con modelos completamente diferentes en su arquitectura (p.ej., no se mantiene el algoritmo de AC) se muestra en la Tabla 2.6. El resultado obtenido por  $BSp_{LLA}$  mejora el resto de los modelos y demuestra que el nuevo modelo LLA puede mejorar la clasificación de aspectos durante el AC de varios dominios.

El modelo presentado en [96], nombrado  $LLM$  en los experimentos, fue superado por el modelo  $BSp_{LLA}$  como consecuencia del uso tanto de BERT como de la estrategia de AC propuesta. Una desventaja durante la experimentación del modelo  $LLM$  y presentada en [96] es que se entrenó solamente en conjuntos de datos de efectos electrodomésticos y no con otros dominios semánticamente diferentes. Esto hubiera permitido apreciar el aporte del  $LLM$  como modelo de AC, al mostrar su efectividad para preservar los patrones comunes entre dominios diversos. En esta experimentación fue evaluado con conjuntos de datos más diversos.

Los resultados de los experimentos muestran que una arquitectura menos compleja como la usada en  $BSp_{LLA}$  obtiene mejores resultados que las que sí lo son (p.ej.,  $AE_{AR1}$ ). La diferencia se centra en que la entrada del modelo BERT es el contexto (palabras en la oración) y el aspecto. Los altos valores de calidad obtenidos se deben a tres características principales: El uso del modelo base BERT en  $BSp_{LLA}$ , su mecanismo de atención con los pesos obtenidos después de ser entrenado en un enorme conjunto de datos [31] y la estrategia de regularización para evitar grandes cambios en los valores de los pesos durante el proceso de AC.

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

Los modelos con la salida de BERT como entrada o como modelos base tienen mejores resultados que aquellos con Word Embeddings. Este resultado es similar al reportado en varios análisis sistemáticos de la literatura [1, 2, 25] y está asociado con la arquitectura que sigue el modelo BERT y su proceso de aprendizaje.

### **2.6.4.1. Prueba de Ablación**

Una prueba de ablación analiza el rendimiento de un modelo de aprendizaje automático a través de la eliminación de uno de sus componentes para determinar la contribución del componente al modelo general. Este tipo de prueba hace una analogía con la biología (e.d., eliminar o remover temporalmente un componente de un organismo), y es empleada principalmente en el análisis de redes neuronales artificiales [152].

En la prueba realizada en esta investigación, fue empleado el *BSp* (e.d., el modelo base de AP) sin el algoritmo LLA (e.d., estrategia de AC o componente que se retira durante la prueba) en el mismo escenario de experimentación que el *BSpLLA*. El objetivo es evaluar el aporte del algoritmo LLA al modelo general, como se muestra en la Tabla 2.7.

<b>Modelo</b>	<i>BSp</i>	<i>BSpLLA</i>
Accr	0.64	<b>0.80</b>
F1	0.52	<b>0.73</b>
Kappa	0.39	<b>0.62</b>
OvrcForgtt	0.52	<b>0.73</b>

Tabla 2.7: Promedio de los resultados de la ablación entre *BSpLLA* y *BSp*

Esta experimentación muestra que el algoritmo *LLA* tiene un influencia importante en los resultados de clasificación y no puede ser eliminado porque provoca pérdida de efectividad.

### **2.6.4.2. Evaluación de la propuesta LLA y una propuesta reciente del estado del arte**

Finalmente, se analizó la propuesta presentada en [2], que constituye una de las propuestas más reciente del estado del arte (e.d., Diciembre, 2021).

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

Modelo	CLASSIC [2]	<i>BSpLLA</i>
Accr	0.90	<b>0.80</b>
F1	0.85	<b>0.73</b>

Tabla 2.8: Resultados de la propuesta de en [2](según artículo) y los de *BSpLLA*.

En [2], se propone un modelo que sigue la estrategia de aprendizaje por contraste [153] y es nombrado CLASSIC, modificando la arquitectura de BERT en dos puntos (e.d., se agregan dos capas de red completamente conectadas) y solo ajustando los pesos de estos nuevos componentes durante el entrenamiento.

Según expresan los autores de dicha propuesta, este modelo tiene un mejor desempeño que LLA (ver Tabla 2.8). Pero al realizar un análisis del mismo y su forma de evaluación se observaron notables diferencias en la manera en que llegaron a los resultados, con respecto a los usados en este trabajo:

- Se realizó la experimentación con 19 conjuntos de datos y esta cantidad incluye 12 más que los usados en nuestra propuesta (e.d., los usados para entrenar *BSpLLA* son parte de los 19).
- Tomaron cinco conjuntos de datos de forma aleatoria para estimar los resultados experimentales. Pero no se definen cuáles fueron, por lo que no se puede establecer un criterio de comparación con los de LLA.
- No se especificó la cantidad de permutaciones posibles entre conjuntos de datos usados para estimar los resultados comparativos del algoritmo propuesto con respecto al estado del arte. La definición de este valor permitiría determinar si el tamaño de la muestra es realmente significativo.
- Se realizaron ajustes diferentes de los parámetros de la red neuronal del modelo computacional según los conjuntos de datos a entrenar (p.ej., la cantidad de 30 ciclos de entrenamientos para los conjuntos de datos de efectos electrodomésticos y 10 ciclos de entrenamiento en el caso de los conjuntos de datos de opiniones de laptops y restaurantes (mucho más numerosos en cantidad de instancias)).

Esto difiere de las condiciones en que fue entrenado y evaluado el modelo computacional LLA donde se usaron los mismos parámetros (p.ej., ciclos de entrenamiento, lote, entre

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

otros) para todos los conjuntos de datos.

- Durante el análisis del código de la implementación de [2], se encontró que usaron una forma de entrada a la arquitectura BERT, distinta a la que se usó en LLA, esta configuración puede influir en los resultados finales.
- Para estimar el olvido catastrófico hicieron uso de la medida propuesta en [146], que difiere a la usada en LLA.
- En la propuesta de [2], no se hace ningún análisis de la cercanía semántica entre los conjuntos de datos. Esto no permite determinar si el aprendizaje y los resultados finales son sobre conjuntos de datos cercanos o no.

Lograr la generalización en modelos de AC es importante, porque uno de los objetivos de este tipo de aprendizaje es que la misma configuración (e.d., hiperparámetros de las redes neuronales) del modelo tenga buenos valores de calidad para todos los dominios. La variación de esta configuración, atendiendo a los conjuntos de datos o dominios por aprender, no aporta al objetivo descrito. Otra desventaja de los cambios de configuración es la necesidad de distinguir el tipo de conjuntos de datos para ajustar la configuración (p.ej., cantidad de ciclos de entrenamiento), debido a que es necesario usar otro modelo, herramienta externa o entrenar el mismo modelo con este propósito también (aumentando el costo computacional en tiempo de entrenamiento y memoria).

A partir de estas diferencias se realizó un análisis comparativo de ambos modelos teniendo en cuenta los siguientes criterios:

- Comparación de ambos modelos (e.d., *CLASSIC* y *BSpLLA*) sobre los mismos conjuntos de datos de entrenamiento y prueba para el AC (los usados en el entrenamiento del modelo LLA que tiene un estudio de su cercanía semántica).
- Uso de la misma medida para estimar el olvido catastrófico (usando la propuesta en [2] porque es muy similar a otras empleadas en trabajos de análisis de sentimientos [25]).
- Uso del mismo conjunto de hiperparámetros para todos los conjuntos de datos como en *BSpLLA*.

Los resultados de los experimentos fueron estimados en base a los valores promediados de la medida F1-macro.

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

Experimento	CLASSIC	$BSp_{LLA}$	Información
<i>Same-phd</i>	0.311	<b>0.316</b>	Los mismos conjuntos de datos para ambos modelos.
<i>Same-parameters</i>	0.182	<b>0.316</b>	Los mismos conjuntos de datos en LLA y en [2], pero con los mismos hiperparámetros (e.d., cantidad de ciclos de entrenamiento igual a 10).
<i>Invert-input</i>	0.311	<b>0.316</b>	Invertir la forma de tokenizar la entrada a BERT con los mismos conjuntos de datos.

Tabla 2.9: Resultados de los experimentos para estimar el mejor desempeño entre [2] y  $BSp_{LLA}$ .

En el experimento con el mismo conjunto de datos (Ver en la Tabla 2.9 los resultados para *Same-phd*) que el modelo propuesto en este trabajo (e.d.,  $BSp_{LLA}$ ), no se obtiene una diferencia significativa (e.d., es de 0.005). Sin embargo, durante este experimento, para el caso de CLASSIC se mantuvo la característica establecida por sus autores de valores de hiperparámetros diferentes para conjuntos de datos (p.ej., el conjunto de datos de dispositivos electrónicos tiene mayor ciclo de entrenamiento y lote). Esto influye en los resultados, porque el modelo puede aprender mejor realizando una búsqueda más extensa de mejores soluciones (según el tipo de conjunto de datos o dominio); pero es una desventaja del modelo con respecto al LLA porque no permite considerar hiperparámetros homogéneos para todos los conjuntos de datos, como se explicó anteriormente.

Las carencias, para reducir el olvido catastrófico, del modelo CLASSIC son demostradas en el experimento donde se mantuvieron los mismos hiperparámetros para todos los conjuntos de datos (Ver en la Tabla 2.9 los resultados para *Same-parameters*). La diferencia entre los resultados de estos modelos es de 0.134 y reafirma la hipótesis de que el modelo CLASSIC no es mejor que el LLA porque no generaliza los hiperparámetros para todos los conjuntos de datos y tampoco hace un ajuste selectivo de la modificación de los pesos de la red durante el AC.

En el experimento donde la forma de representación de los datos de entrada al modelo CLASSIC es transformada para que se corresponda a la usada durante el entrenamiento de LLA (Ver en Tabla 2.9 los resultados para *Invert-input*) se obtiene un resultado similar al experimento

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

nombrado *Same-phd*, porque se mantienen los valores de los hiperparámetros de entrenamiento propuestos en CLASSIC. Este experimento demuestra que esta modificación no tiene un alto impacto en los resultados.

Finalmente, los resultados de los experimentos (Ver Tabla 2.9) muestran que el empleo de un modelo que permita evitar grandes cambios en el valor de los pesos de la red neuronal durante el cálculo del DGE en el proceso de AC, tiene una influencia positiva en los resultados finales. Esta conclusión se puede establecer a partir de analizar la ecuación 2.4 del modelo Inteligencia Sináptica, para la estimación de la pérdida durante el entrenamiento. En esta ecuación, términos como  $\Phi_t$  permiten compensar o evitar grandes cambios en los pesos al actualizar las neuronas durante el DGE. Esta compensación es clave en un proceso de AC, porque permite que el conocimiento anterior no sea completa o parcialmente modificado y ocurran omisiones u olvidos al aparecer instancias diferentes en los nuevos dominios.

El ajuste de los pesos de una parte de la red neuronal BERT (como se propone para CLASSIC) no es mejor que el uso de un modelo de AC basado en regularización, que permite mantener los valores comunes en la última capa asociada al proceso de clasificación de aspectos.

En el caso de [2], se agregan nuevos componentes a la arquitectura de red BERT y esto permite aprovechar las ventajas de esta arquitectura durante el aprendizaje del modelo. También es importante destacar que la actualización de los pesos de la red neuronal solamente en estos nuevos componentes es una disminución del costo computacional en cuanto al tiempo de ejecución en el entrenamiento. Si embargo, no existe en este modelo una compensación o regularización durante el proceso de actualización de los pesos.

### **2.6.5. Análisis de la complejidad computacional del entrenamiento del modelo para la clasificación de aspectos**

Para realizar este análisis se toma como base la complejidad computacional de un algoritmo de AC que emplea como base un modelo de AP descrita en el epígrafe 2.1.2, donde se establece que es  $O(mp\frac{n}{b})$ .

El modelo LLA combina el ajuste de los pesos de un modelo pre-entrenado de tipo BERT que trasforma su salida en la entrada de un modelo de una capa de neuronas completamente conectada para el AC. Este proceso se realiza para  $m$  conjuntos de datos o dominios de entrenamiento.

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

---

En la propuesta de [82] se establece que la complejidad de una capa para un modelo de MA es de  $O(r^2 * d)$  y en el caso de una red recurrente del tipo LSTM es  $O(rd^2)$ . Una capa de MA conecta todas las posiciones con un número constante de operaciones ejecutadas secuencialmente, mientras que una capa recurrente requiere  $O(r)$  operaciones secuenciales.

A partir del análisis de la complejidad computacional anterior, las capas de MA (principal componente de la arquitectura de red neuronal en BERT) son más rápidas que las capas recurrentes cuando la longitud de la secuencia  $r$  es más pequeña que la dimensionalidad de la representación  $d$ , que suele ser el caso en las representaciones de oraciones utilizadas por los modelos diseñados en esta investigación.

En la propuesta de [82] se establece que la complejidad por capas de BERT es  $O(r^2d + rd^2)$ . El pre-modelo de BERT usado en el entrenamiento del modelo LLA tiene 12 capas. En el entrenamiento del modelo, la salida del pre-modelo BERT es la entrada de una capa de tres neuronas completamente conectadas para el AC. A partir de un análisis análogo al de Lwf-CNN-IgR, para una complejidad de  $12 * O(r^2d + rd^2)$  y se reduce a  $O(r^2d + rd^2)$ .

Para el modelo LLA su costo computacional total se define por la ecuación:

$$O(mp \frac{n}{b} (r^2d + rd^2)) \quad (2.22)$$

En las experimentaciones realizadas se compara el LLA con tres modelos de AC (AR1, EWC y CLASSIC) manteniendo el modelo base BERT. Para analizar la complejidad en este caso es importante considerar que AR1 y LLA realiza el ajuste de los pesos para evitar el olvido catastrófico en la última capa que tiene 3 neuronas completamente conectadas a la salida de BERT.

LLA tiene una complejidad computacional mayor que AR1 porque realiza el ajuste de los pesos de la capa del modelo de AC y del modelo BERT, usando el algoritmo SI.

En el caso del EWC, emplea para el cálculo del valor del gradiente y evitar el olvido catastrófico la matriz de Fisher. Esto lo hace mucho más costoso que LLA, porque la complejidad aproximada del cálculo y actualización de esta matriz es de  $O(r^3)$ , pero puede llegar a reducirse a  $O(r^2 \log r)$ , como se propone en [154]. El uso y actualización de la matriz de Fisher genera un costo adicional de memoria porque es un elemento externo a la construcción del modelo que debe ser actualizado después del aprendizaje de cada lote. Sin embargo LLA y EWC, en

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

las experimentaciones, realizan el ajuste de la red neuronal de BERT y el EWC tiene un costo computacional de  $O(r^2d + rd^2 + r^2 \log r)$ , mayor que el de LLA.

La arquitectura de red neuronal en *LLM* es de dos capas de MA y seis capas completamente conectadas, algunas partes de este modelo son paralelizables para la estimación del vector de salida de estas capas, lo que reduce la posible complejidad computacional. Definiendo la complejidad de esta propuesta por  $6 * O(r^2) + 2 * O(r^2 * d)$  donde  $r$  además de ser el tamaño del vector es la cantidad de elementos a procesar en cada capa, siendo su complejidad mayor  $O(r^2 * d)$ . Esto permite definir que este modelo tiene una menor complejidad computacional que LLA (*BSpLLA*).

El modelo CLASSIC añade un componente de dos capas de red neuronal completamente conectadas en dos puntos específicos de la arquitectura de red de BERT, para un total de 4 capas. Durante el entrenamiento solamente son modificados los pesos en estas capas y se mantienen sin modificación los pesos pre-entrenados de BERT. La complejidad computacional para una capa de red completamente conectada es de  $O(r^2)$  [155].

En CLASSIC, la complejidad es de  $4 * O(r^2)$  y generalizando de  $O(r^2)$ . Este modelo tiene una menor complejidad que *LLM*, *BSpLLA* y AR1, aunque no alcanza mejores resultados en su efectividad.

La propuesta LLA es evaluada empleando otros modelos base que no son del tipo BERT : *AE*, *AT*, *LC*. Estos modelos son entrenados de forma secuencial o continuo (no igual a los trabajos donde fueron propuestos).

Con respecto a la propuesta *BSpLLA* (e.d., combina el modelo base BERT con el modelo LLA) la complejidad de *AT* se de  $O(r^2d + rd^2)$  por cada capa, porque combina una red neuronal de MA (usando su salida) con un modelo LSTM. Esta complejidad es similar a la de BERT, pero su eficiencia es mayor en cuando al uso de memoria, con respecto a BERT, porque tiene una menor cantidad de capas.

Aunque *AT* tiene una menor complejidad que *BSpLLA*, su arquitectura de red neuronal no le permite, como en el caso de BERT, aprender de manera simultánea sobre el contexto asociado a varios tokens de la oración y otras características propias del modelo BERT (p.ej., la inserción de forma aleatoria de etiquetas en el texto para aumentar la variabilidad y la eficiencia de la predicción). Estas desventajas reducen su eficacia al ser comparado con BERT.

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

---

El modelo *AE* es una propuesta basada en MA que utiliza como entrada la salida de BERT y tiene una arquitectura de red neuronal que se basa principalmente en dos capas de MA. La complejidad computacional es de  $2 * O(r^2d)$  que es menor a la de *BSpLLA*, pero su arquitectura de red no posee las ventajas descritas anteriormente.

El modelo *LC* propone el uso de dos capas de MA para procesar en paralelo los tokens de una oración y el token asociado a un aspecto y los vectores resultantes son concatenados para ser insertados como entrada a una quinta capa de MA. Este modelo solo emplea a BERT como método de conversión de los *tokens* de una oración en un vector de valores reales (Word Embeddings). El uso de cinco capas de MA y teniendo que el costo computacional de una capa de MA es  $O(r^2d)$ , establece la complejidad computacional de *LC* en  $5 * O(r^2d)$ , al eliminar valores numéricos tenemos  $O(r^2d)$ .

La complejidad computacional de *LC* es menor con respecto a BERT, por tanto la combinación de *LC* como modelo base y de *LLA* como modelo de AC es menor a la de *BSpLLA*.

Se puede concluir que el modelo BERT es más complejo computacionalmente que el resto de los modelos base con los que ha sido comparado, aunque su complejidad también puede reducirse según el tamaño de la secuencia  $r$  y la dimensión del vector  $d$  para los *tokens* que se usan como entrada. Esta desventaja se compensa con la capacidad que tiene la arquitectura de red neuronal para almacenar información, de manera simultánea, sobre el contexto en el que es entrenada [82] y las ventajas del entrenamiento de un modelo AP descritas en el epígrafe 2.1.2.

Para los modelos de AC que son evaluados con respecto al LLA (*BSpLLA*), el de menor complejidad y mayor eficiencia computacional es el LLM; sin embargo al no contar con recursos como el pre-modelo BERT y no enriquecer el proceso de aprendizaje con el ajuste del modelo como lo realiza el *BSpLLA*, tiene bajos valores de eficacia.

## **2.6 Evaluación de la propuesta para la clasificación de aspectos**

### **2.6.6. Evaluación de la estrategia de influencia del orden del aprendizaje de los dominios**

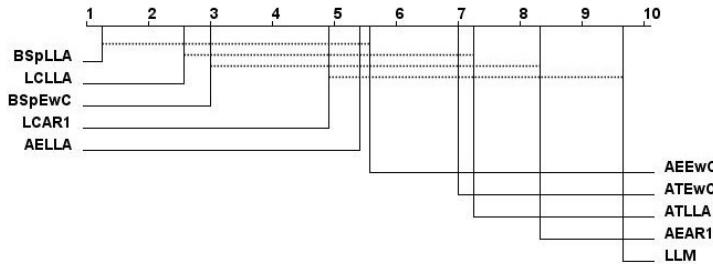


Figura 2.8: Evaluación de los modelos con la prueba de Holm con un nivel de especificidad de 0.05 para la medida de F1-macro.

Las pruebas de Friedman y el método de Holm, para análisis *post hoc* [156, 157], fueron usadas para verificar si existían diferencias significativas entre los modelos con las medidas de desempeño exactitud, Kappa y F1 en las figuras 2.8-2.9. Los experimentos mostraron que LLA no tiene diferencias significativas con los modelos de SOTA evaluados.

Los valores de las medidas F1-macro y Kappa obtenidos para el método propuesto son mejores que los observados en otros métodos incluidos en el estado del arte. Los resultados obtenidos durante el proceso de evaluación validan la nueva propuesta para la subtarea ABSA en un ambiente multidominio.

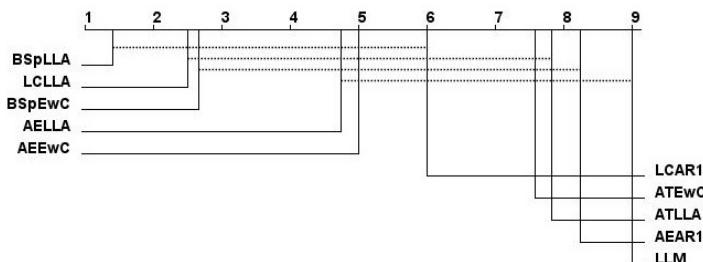


Figura 2.9: Evaluación de los modelos con la prueba de Holm con un nivel de especificidad de 0.05 para la medida Kappa.

## 2.6 Evaluación de la propuesta para la clasificación de aspectos

---

Los resultados de la prueba de Holm no muestran diferencias significativas entre los resultados del LLA y otros de SOTA. Sin embargo, el uso del algoritmo de Inteligencia Sináptica (*Synaptic Intelligence*; SI), adaptado por primera vez en esta investigación al ABSA, tiene menor costo computacional que el EWC [38]. La realización de la actualización de los pesos, en SI, durante el gradiente estocástico descendente y la actualización hacia atrás permite optimizar el tiempo de ejecución del modelo.

Modelo	<i>REH</i>	<i>HER</i>	<i>RHE</i>	<i>HRE</i>	<i>EHR</i>	<i>ERH</i>
<i>BSpLLA</i>	0.70	0.76	0.77	0.78	0.72	0.66

Tabla 2.10: Valores de F1-macro del modelo propuesto para cada ordenamiento de los conjuntos de datos.

La diferencia entre dominios mostrada en el mapa de calor de la figura 2.7 tiene una influencia en la aparición del olvido catastrófico en el modelo propuesto. Como se describe en el análisis experimental, se tomaron seis órdenes de aparición de los conjuntos de entrenamiento (dominios). En la figura 2.10 se muestra el desempeño del modelo propuesto con respecto a cada ordenamiento.

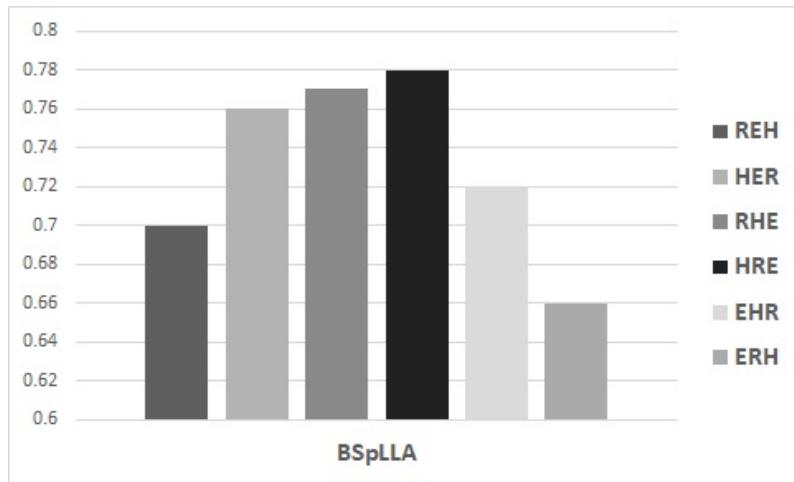


Figura 2.10: Resultados de *BSpLLA* para los seis ordenamientos de los conjuntos de entrenamiento.

En la figura anterior se muestra que se obtienen menores resultados de desempeño siempre que se aprende el dominio de efectos electrodomésticos (e.d., conjuntos de datos de opiniones

sobre laptops, cámaras digitales, speakers, routers) y luego los de hoteles y restaurantes (no importa el orden en que aparezcan).

Aunque el modelo reduce el olvido catastrófico este siempre aparece; sobre todo cuando los conjuntos de datos comienzan a ser diferentes o los ejemplos presentes en ellos son más distantes entre sí. Otra consecuencia del orden de entrenamiento consiste en que el primer dominio entrenado influye en el aprendizaje de los modelos siguientes, porque este es el que fija los valores iniciales de las neuronas (a partir de las modificaciones hechas al algoritmo LLA y tomado de AR1). Los estudios posteriores deben orientarse a la evaluación de otros métodos de inicialización y reducción del olvido catastrófico.

El tiempo de entrenamiento de los métodos basados en BERT fue de 24 horas aproximadamente, es decir, mucho más largo que el obtenido para los métodos basados en Word Embeddings, que fue aproximadamente de tres horas.

Este tiempo está relacionado con la dimensión de la arquitectura y los MA de BERT usados en el procesos de aprendizaje. Por ejemplo, el premodelo BERT tiene 110 millones de parámetros o pesos; durante el cálculo del DGE y la propagación hacia atrás se debe aplicar el algoritmo SI y ajustar estos parámetros, aumentando el tiempo de ejecución.

## 2.7. Conclusiones parciales

En el capítulo se presentan dos nuevos modelos: uno para la extracción de aspectos nombrado Learning without Forgetting with Linguistic Rules (Lwf-CNN-IgR) y otro para la clasificación de aspectos, nombrado Lifelong Learning of Aspects (LLA).

El primero de ellos combina una red neuronal convolucional y un model de AC para proponer un nuevo marco de trabajo que, aunque fue basado en el propuesto por Poria et al para la extracción de aspectos sobre un único dominio, fue diseñado para trabajar sobre varios dominios cercanos entre sí. El nuevo modelo propuesto mantiene la red neuronal de siete capas del original y respeta los valores de los parámetros propuestos por sus autores.

Como aporte importante del modelo, se definió un algoritmo que tiene como entrada un vector que representa los aspectos presentes en las oraciones del conjunto de datos y la posible clasificación de los mismos, y como salida los pesos de las neuronas que serán usados en el modelo base CNN y en la capa de clasificación del modelo Lwf.

## 2.7 Conclusiones parciales

---

El modelo propuesto fue evaluado tanto para los dominios anteriores como para el actual, obteniéndose un mejor comportamiento con respecto a otros del estado del arte, al obtener un valor de F1-macro cercano a 0.8 para el dominio cruzado, y uno superior, o similar al resto de las propuestas, cuando se trata del dominio actual (entre 0.78 y 0.81). Esto se debe fundamentalmente a la compensación o penalización de la modificación de los pesos de las neuronas que establece la regularización en la función de costo del modelo propuesto, con respecto a los resultados obtenidos en los dominios anteriores.

Sin embargo, cuando se realizó un segundo conjunto de pruebas, sobre un esquema de dominio cruzado, agregando nuevos dominios distantes entre sí, se evidenció que la propuesta es aun sensible a la diversidad de dominios, lo que provoca la aparición del olvido catastrófico. Es por ello que se recomienda su uso para conjuntos de dominios semánticamente cercanos entre sí.

Otra conclusión importante es que el uso de las reglas lingüísticas permite incrementar el conjunto de datos de entrenamiento, logrando mejorar los resultados de métodos de AP, cuando este conjunto es pequeño. Esto se debe principalmente a que permiten aumentar el conjunto de datos para un posible entrenamiento de un clasificador y al mismo tiempo descubrir nuevas reglas asociadas al dominio de aprendizaje.

La segunda propuesta de este capítulo consiste en un nuevo método de clasificación de aspectos que usa el AC en la subtarea ABSA, resultado de la adaptación de otra propuesta usada para la clasificación de imágenes REF. Este modelo hace uso de BERT como método de representación del lenguaje natural en varias de sus etapas, lo que mejora los resultados con respecto a otras formas de representación como Word Embeddings. Las adaptaciones que le fueron realizadas evitan que los pesos de las neuronas de la arquitectura de red tengan grandes variaciones durante el proceso de aprendizaje continuo de nuevos dominios. Esto permite reducir los efectos del olvido catastrófico.

Como parte de este modelo se definió un segundo algoritmo que como salida obtiene tanto el vector de pesos usado en la inferencia como el conjunto de pesos del modelo base. Para evaluar el comportamiento de este segundo modelo se usaron las mismas medidas de calidad que en el primer caso.

Para la estimación del desempeño y el olvido catastrófico se tuvieron en cuenta un conjunto de medidas de la literatura, seleccionándose finalmente las medidas Kappa, por ser útil para de-

## 2.7 Conclusiones parciales

---

terminar la eficacia del modelo con datos desbalanceados. OvrcForgtt, por permitir estimar la disminución del desempeño del modelo durante su proceso de aprendizaje secuencial de varios conjuntos de datos y haber sido empleada en otros trabajos sobre análisis de sentimientos.

Para entrenar y evaluar el desempeño de esta propuesta con respecto a las principales del estado del arte, los experimentos se realizaron empleando siete de los conjuntos de datos más usados en ABSA, cercanos semánticamente entre sí, considerando distintos órdenes de entrenamiento, para demostrar la inmunidad de la propuesta a estos cambios, obteniéndose un comportamiento similar en todos los casos. Los mejores valores obtenidos para las medidas de calidad usadas fueron  $F1\text{-macro} = 0.73$ ,  $Kappa = 0.62$ ,  $OvrcForgtt = 0.316$ , siendo superiores a las del resto de los métodos comparados.

El modelo BERT empleado como modelo base de LLA es más complejo computacionalmente que el resto de los modelos con los que ha sido comparado. Para reducir su complejidad es posible reducir el tamaño de la secuencia o vector de entrada a cada capa de la red y la dimensión de este vector. Esta desventaja se compensa con la capacidad de BERT de almacenar información del contexto y el uso de recursos computacionales de alto desempeño para su entrenamiento.

De los modelos de AC comparados con respecto a LLA ( $BSp_{LLA}$ ), el LLM y CLASSIC se destacan por tener menor complejidad y mayor eficiencia computacional que LLA pero los resultados de eficacia y el ajuste del modelo BERT durante el entrenamiento de conjunto con el SI ratifican la selección de LLA.

Los resultados de los experimentos muestran que una arquitectura menos compleja como la usada en  $BSp_{LLA}$  obtiene mejores resultados que otras de mayor complejidad (p.ej.,  $AE_{AR1}$ ). La diferencia se centra en que la entrada del modelo BERT es el contexto (palabras en la oración) y el aspecto. Los altos valores de calidad obtenidos se deben a tres características principales: El uso del modelo base BERT en  $BSp_{LLA}$ , su mecanismo de atención con los pesos obtenidos después de ser entrenado en un enorme conjunto de datos [31] y la estrategia de regularización para evitar grandes cambios en los valores de los pesos durante el proceso de AC.

Finalmente, al comparar este modelo con una propuesta más recientes del estado del arte y que prometía mejores resultados, se comprobó que al generalizar esta última surgieron determinadas insuficiencias que provocaron resultados inferiores a los del modelo propuesto cuando se hace uso de la medida de calidad  $F1\text{-macro}$ , sobre todo cuando se evalúan ambas propuestas

## **2.7 Conclusiones parciales**

---

con los mismos valores de hiperparámetros.

Esto se debe fundamentalmente a que el modelo propuesto en esta investigación permite evitar grandes cambios en el valor de los pesos de la red neuronal durante el cálculo del descenso del gradiente en el proceso a partir del uso del algoritmo de Inteligencia Sináptica compensando los cambios de los pesos actualizar las neuronas. Esta compensación es clave en un proceso de AC, porque permite que el conocimiento anterior no sea modificado y ocurran omisiones u olvidos al aparecer instancias diferentes en los nuevos dominios.

Se mostró la infuencia del ajuste de los pesos en toda la arquitectura neuronal del modelo base BERT sobre la calidad final de los resultados, al lograrse mejores valores de F1-macro, Kappa y OvrcForgtt durante la comparación de la propuesta con otras de SOTA.

Estos resultados permiten concluir que las propuestas presentadas en esta investigación son capaces de obtener altos resultados de calidad durante el proceso de clasificación de opiniones textuales, con una reducción mayor del olvido catastrófico que otras propuestas del estado del arte. En el capítulo siguiente se abordará el uso de los modelos propuestos en importantes servicios de la sociedad.

## **CAPÍTULO 3**

### **El análisis de sentimientos basado en aspectos para la gestión de campañas de bien público**

La información que pueden obtener los seres humanos aumenta constantemente [4]; las vías de comunicación para el acceso a ella también son muy diversas (libros, radio, televisión, internet, etc.). Una tarea importante para la sociedad es la diseminación de la información y el conocimiento para mejorar estándares de vida o enfrentar grandes retos (cambio climático, racismo, enfermedades de trasmisión sexual o altamente contagiosas, etc) [158, 159]. Conocer las opiniones sobre estos problemas y poder mejorar el acceso y la calidad de la información es importante para las organizaciones sociales y otros actores de la sociedad, porque permiten obtener mejores resultados en la gestión de sus actividades y el mejoramiento de la sociedad.

En este capítulo se comentará sobre las campañas sociales o de bien público. Se abordará el problema del procesamiento y análisis de la retroalimentación a través de las opiniones de estas campañas, como una herramienta importante para la toma de decisiones. Se mostrará la plataforma WisePocket para la gestión digital de campañas de bien público y el uso en ella de los resultados de esta investigación. Se presenta un esquema general de aplicación de los modelos computacionales para el ABSA [4] creados en esta investigación y su representación textual.

#### **3.1. La gestión de la información en las campañas de bien público**

Las campañas sociales o de bien público son una herramienta importante para promover un cambio positivo en las actitudes sociales (p.ej., en ecología, prevención de la salud, promoción de la tolerancia, etc.). Aumentar su efectividad puede tener un resultado tangible en muchos aspectos de la vida, tanto para los individuos como para las sociedades. Entre las actividades más extendidas que se llevan a cabo en una campaña social se encuentran las alertas a través de diferentes tipos de medios: televisión, radio, internet y documentos impresos. La evaluación de estos elementos se realiza principalmente a través de cuestionarios y en grupos de personas [158, 159, 160].

### **3.1 La gestión de la información en las campañas de bien público**

---

En ambientes rurales obtener información sobre la salud, el gobierno y otros tópicos es un reto y en ocasiones resulta costoso por el poco alcance que pueden tener los medios de difusión masiva en estos contextos. Los medios tradicionales como la prensa, la radio y la televisión tienen un duración definida en el tiempo (p.ej., hora de emisión de noticieros, programas de radio) y ofrecen poca garantía de que la nueva información sea obtenida o retenida por la población objetivo [161].

Para superar estos retos, las organizaciones sociales emplean las nuevas Tecnologías de la Informática y las Comunicaciones (TICs) en sus campañas de bien público. El uso de tecnologías TICs, como los teléfonos inteligentes, ha sido parte de las estrategias de divulgación de la información de estas organizaciones y para el aumento del alcance de su público objetivo [162].

#### **3.1.1. Procesamiento de la información en las campañas de bien público: Herramientas digitales**

Estas campañas emplean las TIC como las redes sociales o blogs para la divulgación de la información. En estos medios el público objetivo de las campañas puede comentar, enviar datos, o sugerir nuevas propuestas. El uso de estas vías de comunicación generan una cantidad de información que hace difícil, en un tiempo aceptable, por el personal que atiende las campañas, encontrar datos relevantes sin perder otros.

Una forma tradicional para obtener retroalimentación de los resultados de estas campañas es la realización de encuestas de opinión a una muestra representativa del público objetivo de la campaña [163, 164]. Su realización sigue el formato tradicional en medios impresos y son analizadas por un equipo de expertos [164].

El uso de herramientas digitales permite humanizar el trabajo y obtener resultados en un corto tiempo. La información a procesar en su mayoría es textual (opiniones o comentarios, cartas al personal de la campaña, entrevistas o resultados de las encuestas) [163]. El uso de estas herramientas se convierte en una opción muy empleada por las organizaciones para obtener los diferentes tipos de retroalimentación en las campañas [161, 162].

### **3.1 La gestión de la información en las campañas de bien público**

---

#### **3.1.2. El análisis de sentimientos basado en aspectos para el procesamiento de la información**

El uso del análisis de sentimientos para el procesamiento de la información de los resultados de una campaña se puede realizar sobre información obtenida a través de diversas fuentes. Una de ellas pueden ser los pequeños comentarios o *tweets* publicados sobre estas campañas en Twitter<sup>1</sup> [165]. El uso del ABSA permite una mayor información porque puede caracterizar mejor las opiniones o recomendaciones del público de una campaña. Esta subtarea del análisis de sentimientos puede obtener más información sobre las entidades, estableciendo una clasificación (p.ej., positiva, negativa o neutra), según el criterio expresado por los usuarios y la predicción de modelos computacionales empleados para extraer y clasificar la información de forma automática.

La recuperación de la información para el análisis de sentimientos orientado al ABSA en las campañas de bien público requiere la integración de varias áreas del PLN: el pre-procesamiento de la información, el reconocimiento de nombre de entidades, el uso de modelos de lenguajes como Word Embedding o BERT, la clasificación del sentimiento mediante modelos computacionales y la visualización de la información [4, 50]. La clasificación del sentimiento permite determinar la información relevante y crear un nuevo conocimiento a partir de estos datos, al permitir la toma de decisiones sobre lo que gusta, molesta o valora el público objetivo de las campañas.

Un ejemplo del uso de herramientas para el ABSA en el flujo de opiniones, es en la toma de decisiones de importantes empresas de productos y servicios digitales [4, 80, 97].

Estas herramientas se pueden dividir en dos grupos:

- Interfaces para el desarrollo de aplicaciones a través de la consulta de servicios (Servicios Web) desde sistemas de terceros.
- Plataformas o sistemas que realizan el procesamiento de diversas fuentes de información y ofrecen un reporte a sus usuarios o clientes.

La empresa Clarabridge<sup>2</sup> ofrece una herramienta para el análisis de las experiencias de usuarios hacia productos y servicios, permitiendo realizar el análisis ABSA basado en el contexto

---

<sup>1</sup><https://twitter.com>

<sup>2</sup><https://www.clarabridge.com/>

### **3.1 La gestión de la información en las campañas de bien público**

---

de fuentes como redes sociales, correos, entre otras. Otro producto es el de la empresa Repus-tate que procesa el flujo de información desde diversas fuentes como Clarabridge y permite el análisis en 17 idiomas.

La plataforma OpenText Analytics<sup>3</sup> tiene una herramienta para el análisis de sentimientos, que toma tres niveles diferentes: aspectos, oraciones y documentos. Permite evaluar si una pieza determinada del contenido es positiva, negativa, neutral o tiene varias polaridades. Lexalytics<sup>4</sup> ofrece una herramienta que realiza el análisis de sentimientos, permitiendo la categorización, el reconocimiento de nombres de entidades, la detección de la intención. Usa un sistema de análisis de sentimientos híbrido, que combina modelos de aprendizaje automatizado y reglas lingüísticas. El sistema no asigna solamente una puntuación al documento, también a las entidades individuales, tópicos, temas y categorías.

Otras plataformas como Meaning Cloud<sup>5</sup> proponen una interfaz para la programación de aplicaciones (API: Siglas en inglés). Esta API permite realizar un análisis de ABSA al identificar la polaridad local de las diferentes frases en el texto y se evalúa la relación entre ellas, lo que resulta en un valor de polaridad global para el texto en su conjunto. El API convierte a los servicios en una herramienta aplicable a cualquier tipo de escenario.

La plataforma MonkeyLearn<sup>6</sup> permite realizar todos los pasos para la creación de modelos de ABSA a la medida del proyecto o solución que se diseña. Ofrece una herramienta de análisis de sentimientos fácilmente configurable. Permite la creación de etiquetas de categorización y la selección de diferentes partes del texto para mostrar el contenido perteneciente a la etiqueta. Permite el entrenamiento de los modelos que ofrece a las soluciones de terceros de forma semi-supervisada, permitiendo ajustar el aprendizaje. Ofrece además un API de desarrollo para la creación de aplicaciones de terceros. Microsoft Text Analytics API<sup>7</sup> brinda como una de sus funcionalidades el análisis de sentimientos a partir de la versión 3.1. Esta funcionalidad realiza el ABSA y proporciona información más granular sobre las opiniones relacionadas con aspectos (como los atributos de productos o servicios) en el texto.

---

<sup>3</sup><https://www.opentext.com>

<sup>4</sup><https://www.lexalytics.com/>

<sup>5</sup><https://www.meaningcloud.com/products/sentiment-analysis>

<sup>6</sup><https://monkeylearn.com/blog/aspect-based-sentiment-analysis/>

<sup>7</sup><https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis>

### **3.1 La gestión de la información en las campañas de bien público**

---

La herramienta Aspect-based-sentiment-analysis en su versión 2.0.2<sup>8</sup> es un módulo de python y un proyecto público para el ABSA, que reúne los modelos de AP propuestos en varias investigaciones y permite el entrenamiento de estos modelos con nuevos conjuntos de datos o modificar el código fuente. En la propuesta ABSA-Toolkit<sup>9</sup> se tiene un conjunto de herramientas para ABSA en las opiniones de los clientes. El sistema tiene dos fases principales: de desarrollo y de producción. La fase de desarrollo permite al usuario entrenar modelos para realizar tareas de ABSA en el dominio de destino. En la fase de producción se genera una aplicación web a través de la cual el usuario final puede enviar reseñas para analizar los sentimientos a nivel de aspecto.

La mayoría de los sistemas mencionados están dirigidos al procesamiento de información de productos, servicios desde diversas fuentes, como redes sociales, blogs, periódicos digitales y otras en Internet. El empleo de los servicios para el análisis de sentimientos en estos sistemas tiene un alto precio en función de la cantidad de información y el tiempo de uso. Su utilización por instituciones u organizaciones cubanas se limita en algunos casos el acceso por el bloqueo económico a Cuba de los Estados Unidos. Otras propuestas (p.ej., Aspect-based-sentiment-analysis), que son de acceso abierto, fueron creadas a partir de modelos computacionales que no incluyen el AC, y presentan dificultades en la calidad de los resultados en flujos de opiniones de diversos dominios.

En Cuba algunas plataformas como Lidt-Noti de la empresa DATyS<sup>10</sup> permiten el análisis de sentimientos en idioma español, pero no a nivel de aspecto (ABSA). Durante el transcurso de esta investigación no se pudo identificar en Cuba una propuesta o solución que permita el análisis de ABSA en idioma inglés para el análisis de la información en campañas de bien público, promoción de actividades o instituciones o que apoye la gestión del turismo, la cultura o la salud.

En este trabajo se propone un esquema general para la aplicación de la extracción y clasificación de sentimientos basado en aspectos (ABSA) en el procesamiento y recuperación de la información digital creada por las opiniones de usuarios de las campañas de bien público. Este esquema general contribuye a la gestión de la información y el conocimiento de las organizaciones que realizan estas campañas.

---

<sup>8</sup><https://pypi.org/project/aspect-based-sentiment-analysis/>

<sup>9</sup><https://github.com/zarmeem92/ABSA-Toolkit>

<sup>10</sup><http://www.datys.cu/spa/site/product/14>

### **3.2 WisePocket: Plataforma digital para campañas de bien público**

---

La fuente de información de este esquema proviene de diversas vías (internet, correo electrónico, el servicio de mensajes cortos (SMS:siglas en inglés)) y es almacenada en un servidor, en la colección asociada a cada campaña. La salida son las palabras que representan aspectos o características y su polaridad asociadas a las diferentes entidades en las oraciones presentes en las opiniones. En el Anexo E se muestran los tres módulos que conforman el esquema propuesto:

1. Preprocesamiento de la información.
2. Clasificación de la información.
3. Recuperación y visualización de la información.

El sistema desarrollado, que aplica este esquema general, es el módulo para el ABSA de la plataforma WisePocket.

## **3.2. WisePocket: Plataforma digital para campañas de bien público**

La plataforma WisePocket es un conjunto de aplicaciones (Generador de Contenido, Aplicación para dispositivos móviles y Aplicación web para la recepción de información) que permiten a las instituciones de salud, educación y no gubernamentales la construcción del conocimiento para mostrar al público en campañas sociales y la retroalimentación de las posibles opiniones y preguntas. Está compuesta por una arquitectura cliente-servidor, servicios para la creación dinámica de contenido y la integración con herramientas para la indexación y análisis automático de opiniones y asistencia virtual (chatbot). Las herramientas y marcos de trabajo empleados en el desarrollo de esta propuesta están soportados sobre licencias de software libre [162].

Los servicios que propone la plataforma, en un marco de trabajo configurable, permiten crear y desplegar rápidamente aplicaciones sin conectividad a redes externas.

Un caso de uso es el empleo de WisePocket en el apoyo a las actividades de promoción del Centro Promotor de Salud del Ministerio de Salud Pública de la República de Cuba, realiza anualmente varias campañas de bien público (p. ej., contra la infestación del mosquito Aedes Aegypti, contra las enfermedades de trasmisión sexual, entre otras.). Instituciones como esta

### 3.2 WisePocket: Plataforma digital para campañas de bien público

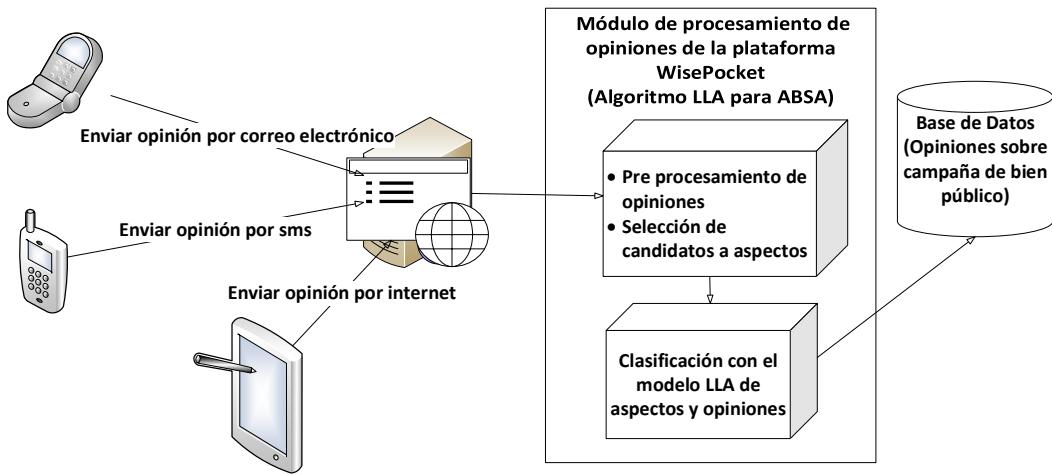


Figura 3.1: Diagrama de despliegue donde se muestra el uso de los modelos *Learning without Forgetting with Linguistic Rules*; Lwf-CNN-lgR) y (*Lifelong Learning of Aspects*; LLA) en la plataforma WisePocket para la retroalimentación de las opiniones de una campaña de bien público.

durante las campañas de bien público siguen un patrón básico: la entrega de documentos (plegables, entre otros), la presentación de conferencias de información o charlas educativas, y la obtención de retroalimentación de sus campañas a través de encuestas en papel o el boca a boca.

Con frecuencia, las personas necesitan más información acerca de estas campañas y surgen varios inconvenientes como: no se cuenta con los recursos suficientes para disponer de un personal humano o digital que dé respuestas, el horario de la atención a usuarios no es 24 horas en la mayoría de los casos; no se conoce, en el momento, la respuesta y no hay forma de retroalimentar en el tiempo al usuario, entre otros inconvenientes. Resulta costosa la creación de una aplicación para dispositivos teléfonos inteligentes para cada campaña de bien público. El costo implica a las organizaciones y a los usuarios que deberían instalar nuevas aplicaciones. La plataforma WisePocket ofrece herramientas para disminuir estas desventajas a través de sus componentes [162].

Esta plataforma obtuvo el premio internacional “Para el Desarrollo Digital”, de la Agencia para el Desarrollo del Reino de Bélgica en 2018 como mejor idea de emprendimiento (iStarUp). En la provincia de Santiago de Cuba ha sido empleada por el Ministerio de Salud Pública durante las epidemias de Dengue, Chika y Chinkunguya en 2018 y durante la epidemia de Covid-19

### **3.3 Módulo para el análisis de sentimientos en WisePocket**

---

en 2020 [162]. La aceptación de funcionarios, personal de la salud y público en general fue muy buena, como lo avalan los criterios recogidos en una encuesta realizada.

#### **3.3. Módulo para el análisis de sentimientos en WisePocket**

Los usuarios que instalan la aplicación móvil de WisePocket pueden enviar opiniones a través de correo electrónico, el servicio de mensajes cortos (SMS) o mediante el uso de internet a los servidores donde se almacena esta información no estructurada para su posterior análisis.

A partir de la realización de tareas periódicas, el módulo de análisis de sentimientos recupera y clasifica las opiniones de una campaña; el resultado se almacena, para la consulta y toma decisiones del personal de una institución que realiza una campaña de bien público.

La construcción del módulo se apoyó en el uso de proyectos y bibliotecas de código abierto disponibles en repositorios como PyPi<sup>11</sup> y GitHub<sup>12</sup>, mantenidos por la comunidad de programadores. El módulo spaCy es la principal herramienta empleada para el PLN. Esta contiene un conjunto de bibliotecas, creadas en lenguaje Python, que ofrecen facilidades para la lematización, obtención de la raíz de palabras, eliminación de palabras de parada, división de oraciones y análisis de dependencias de las oraciones. En proyectos de producción ha sido demostrada su alta eficacia [166], por contar con las funcionalidades necesarias para ser empleada en la minería de datos y el procesamiento de textos.

Para la persistencia de la información se empleó el servidor de base de datos, No-SQL MongoDB<sup>13</sup>. Su empleo se debe a que su forma de organización de la información es orientada a documentos [167]. Esto permite una mayor flexibilidad en los momentos de recuperación y almacenamiento de la información estructurada, porque posibilita adaptarse de forma dinámica a nuevos requerimientos o datos.

El entrenamiento y creación de los modelos de extracción y clasificación de ABSA se realizó usando PyTorch<sup>14</sup> en su versión 1.8.0. Este marco de trabajo permite la creación de modelos de AP de forma eficiente y rápida proporcionando diversas clases y funcionalidades en el desarrollo de los modelos.

---

<sup>11</sup><https://pypi.org/>

<sup>12</sup><https://github.com/>

<sup>13</sup><https://www.mongodb.com/>

<sup>14</sup><https://pytorch.org/>

### **3.3 Módulo para el análisis de sentimientos en WisePocket**

---

El módulo para la clasificación de ABSA es usado mediante el empleo de servicios web de Transferencia de Estado Representacional (*Representational State Transfer*; REST), desarrollados con el uso del marco de trabajo Flask<sup>15</sup>, porque permite una interfaz con un servidor web y posibilita la creación de servicios web y arquitecturas de microservicios de forma ágil [168].

En el módulo para la clasificación de aspectos, al recibir el texto de una opinión, se determina el idioma; si no es en inglés no se realiza ningún tipo de procesamiento. Si lo es esta es pre-procesada a una forma de representación textual que permite su uso como entrada por el extracto y clasificador de aspectos en ABSA propuesto en este trabajo. La forma de representación textual es muy importante para lograr el éxito en el proceso de clasificación de PLN, porque determina la entrada de información de los modelos computacionales y la posterior predicción de los aspectos en las oraciones.

En el Anexo G se muestra la descripción de la representación computacional del texto digital de las opiniones, que permite el procesamiento de la información por los modelos computacionales propuestos en la investigación.

#### **3.3.1. Caso de uso: Campaña de bien público para la divulgación del XIII Seminario Nacional sobre Estudios Canadienses**

Los modelos computacionales propuestos en esta investigación se utilizaron en el XIII Seminario Nacional sobre Estudios Canadienses el 21 de abril de 2022, como parte del módulo de análisis de sentimientos de la plataforma WisePocket. Este evento sesiona en los idiomas inglés, francés y español; participan académicos e investigadores de varias universidades de Cuba y Canadá, es organizado desde la Facultad de Lenguas Extranjeras de la Universidad de Oriente, donde radica la Cátedra Honorífica sobre Estudios Canadienses. El uso de la plataforma contribuyó a la estrategia comunicativa y de imagen de la cátedra, abarcando cuatro componentes principales: funcional, cultural, comercial y visual.

Para lograr los objetivos de divulgación de este evento se usaron las herramientas digitales de la plataforma WisePocket. La aplicación para dispositivos móviles permitió la visualización de información multimedia (en inglés y español) sobre los diversos valores para rescatar y conservar: la labor formativa y educativa, y en consecuencia, con los procesos de gestión de la identidad y la imagen de la Cátedra y de la Universidad de Oriente, su pertinencia y

---

<sup>15</sup><https://flask.palletsprojects.com/en/2.1.x/>

### **3.3 Módulo para el análisis de sentimientos en WisePocket**

---

aplicabilidad para asumir con éxito la misión sociocultural actual.

El módulo de procesamiento de opiniones y la aplicación web para la visualización de la información de la campaña fueron usados por los organizadores para la retroalimentación sobre los principales criterios u opiniones emitidos de los participantes en el evento. Los modelos computacionales para el procesamiento de texto fueron usaron en idioma inglés, logrando procesar y clasificar los aspectos de unas cincuenta opiniones. Se detectaron 120 aspectos (cincuenta positivos, treinta negativos y cuarenta neutrales). Los organizadores, considerando la importancia de la plataforma propuesta y los resultados logrados por la misma, proponen disponer de su uso en futuras versiones del evento y en las actividades nacionales e internacionales de la cátedra.

#### **3.3.2. Otros escenarios en Cuba para el uso de los resultados de la investigación**

Existen diversas áreas de la sociedad cubana donde el empleo de los resultados de esta investigación pudieran ser valiosos. Uno de ellas es la actividad turística. En Cuba se recibieron en 2019 más de 4 millones de turistas<sup>16</sup>.

El país de procedencia de mayor flujo es Canadá, donde el idioma inglés es preponderante. El Ministerio de Turismo y las instalaciones hoteleras no poseen herramientas para el procesamiento de las opiniones de la gran cantidad de usuarios extranjeros.

La herramienta para el PLN en idioma inglés, propuesta en esta investigación, permitiría lograr el análisis de la información asociada a las opiniones en idioma inglés que los turistas emiten sobre las instalaciones hoteleras y otros lugares de esparcimiento, y la toma oportuna de decisiones por el personal de este sector. Se conocen otras investigaciones como la propuesta en [169], donde se aborda el tema.

Otro escenario es el análisis de noticias por parte de las instituciones de defensa del país o donde su objeto social se relacione con el procesamiento, análisis de información de fuentes internacionales en idioma inglés (p.ej., Centro de Prensa Internacional, órganos de análisis de información del Ministerio del Interior). Estas intituciones tienen como fuentes de información importantes periódicos, noticieros y agencias cablegráficas (p.ej., *British Broadcasting*

---

<sup>16</sup>Se muestran datos de 2019 porque la pandemia de Covid-19 en los años 2020 y 2021 redujo la llegada de turistas al país, aunque se incrementa paulatinamente

### **3.3 Módulo para el análisis de sentimientos en WisePocket**

---

*Corporation (BBC)<sup>17</sup>, Cable News Network (CNN)<sup>18</sup>, Associated Press; (AP)<sup>19</sup>.* Los modelos propuestos en esta investigación, de conjunto con otras herramientas de PLN, permitirían el análisis, resumen y descubrimiento de conocimiento de la información, humanizando el trabajo y reduciendo los costos de personal o la compra de otros software para realizar la tarea.

La coordinación y la disposición de los recursos de infraestructura tecnológica necesarios permitirían el uso de los resultados propuestos en este trabajo en los escenarios descritos.

#### **3.3.3. Consideraciones necesarias para extender los modelos propuestos al idioma español**

Los modelos propuestos en esta investigación están orientados al trabajo con información en idioma inglés. Sería deseable poder realizar el entrenamiento de estos modelos para entornos y dominios donde el flujo de opiniones sea en idioma español, teniendo en cuenta la gran cantidad de habitantes en Hispano-América y la gran popularidad e importancia que tiene este idioma. Se expondrán a continuación algunas consideraciones necesarias para lograr este objetivo y que están relacionadas con las experiencias obtenidas en esta investigación.

Una de las más importantes tareas es lograr disponer de un conjunto de datos adecuado para poder entrenar los modelos. Para lograr extender el uso del modelo propuesto en esta investigación al idioma español se debe coleccionar conjuntos de datos sobre opiniones de diversos dominios en este idioma.

Es importante tener en cuenta la cercanía semántica entre estos dominios, por lo que se sugiere realizar las experimentaciones propuestas en el epígrafe 2.5.1 del Capítulo 2. Esto permitirá tener una estimación “a priori” de la posible existencia de patrones comunes entre dominios.

En el idioma español, según nuestro conocimiento, no existen muchos conjuntos de opiniones para el análisis de sentimientos orientado a aspectos. Uno de los posibles candidatos es el propuesto para la competición SemEval 2016 [49] sobre opiniones en dominios de restaurantes y laptops, con similares características a los empleados en esta investigación para el idioma inglés. Es necesario contar con otros conjuntos de datos, y no solo con estos dos porque para realizar el entrenamiento se estaría enfrentando a una tarea de Aprendizaje por Transferencia

---

<sup>17</sup><https://www.bbc.com/>

<sup>18</sup><https://edition.cnn.com/>

<sup>19</sup><https://apnews.com/>

### **3.3 Módulo para el análisis de sentimientos en WisePocket**

---

(e.d., de dominio fuente - a dominio destino).

Otra estrategia para lograr conjuntos de datos sobre dominios más diversos es la creación, por parte del investigador, de un conjunto de datos o mediante la colaboración con otros centros de investigación o servicios donde se procesen flujos de opiniones. Por ejemplo, durante el trascurso de esta investigación se contactó con los directivos de la sección “Cartas al Director” del periódico *Granma*<sup>20</sup> órgano oficial del Partido Comunista de la República de Cuba. En este departamento se reciben opiniones de todo el país, sobre servicios e instituciones de la República de Cuba y se realiza una clasificación de estas en cuanto a polaridad y sectores. Aunque no se logró una colaboración con el periódico, es una muestra de que la colaboración permitiría la construcción de conjuntos de datos a la medida.

Otra institución que se contactó fue la Delegación Provincial de Turismo de Santiago de Cuba, con el objetivo de que se emplearan los resultados de la investigación y con la posibilidad de recopilar conjuntos de datos sobre opiniones en hoteles con idioma español. Esta institución tampoco mostró interés de colaboración, pero pudiera servir como ejemplo a otros investigadores.

Para la construcción de un conjunto de datos, por parte de investigadores o colaboradores, se sugiere que esté avalado por la comunidad científica y que siga las normas para la construcción de este tipo de recursos de PLN [170]. No seguir estas pautas y no obtener el aval de la comunidad científica (p.ej., publicación en revista de impacto, participación en congreso, la certificación de una institución reconocida intencionalmente, entre otras) comprometería la validación de los posibles resultados de calidad obtenidos por los modelos computacionales en el idioma español.

Otro de los elementos importantes para el entrenamiento de los modelos propuestos es un pre-modelo con la arquitectura BERT entrenado para el idioma español. Existen varias propuestas que pudieran emplearse, como:

- BETO: Spanish BERT<sup>21</sup>.
- RoBERTa base entrenado con datos de la Biblioteca Nacional de España (BNE)<sup>22</sup>.

---

<sup>20</sup><https://www.granma.cu/>

<sup>21</sup><https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased>

<sup>22</sup><https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne>

- XLM-RoBERTa (base-sized model)<sup>23</sup>.

Otros recursos pueden ser encontrados en la plataforma [huggingface](#)<sup>24</sup>.

Una desventaja con respecto a los pre-modelos de BERT para el idioma inglés, es que los recursos anteriormente relacionados para el español no fueron entrenados con la gran cantidad de datos con que lo fueron los de inglés. Esto provoca que los resultados obtenidos para el español no tengan la misma variabilidad con respecto al espacio de posibles soluciones que para el de idioma inglés.

Una forma de superar esta desventaja consiste en colecciónar gran cantidad de textos digitales (p.ej., información en bibliotecas nacionales) en idioma español y entrenar el modelo BERT para la construcción de un pre-modelo a partir del ajuste de uno existente. Para lograr esto es necesario contar con los recursos de cómputo (computación de altas prestaciones), tiempo y personal.

Otra de las propuestas para superar los escasos recursos para el idioma español es la combinación de un modelo entrenado de forma supervisada con el uso de reglas lingüísticas y otros recursos para minar conjuntos de datos como las reglas propuestas en [171].

Se considera que estos pasos previos para poder entrenar y evaluar los modelos propuestos en esta investigación, permitirán obtener una versión para idioma español. Con este objetivo se orientarán varias de las investigaciones futuras que se derivan de los resultados obtenidos en este trabajo.

## 3.4. Conclusiones parciales

En este capítulo se muestra, como parte de la plataforma WisePocket, un esquema general que propone la integración de los modelos computacionales para la extracción de aspectos (*Learning without Forgetting with Linguistic Rules*; Lwf-CNN-IgR) y para la clasificación de aspectos (*Lifelong Learning of Aspects*; LLA) en ABSA.

Estos modelos muestran su flexibilidad y facilidad como herramientas de procesamiento del lenguaje natural en el procesamiento de opiniones en idioma inglés de campañas de bien público. Los datos mostrados en este capítulo sobre el uso de estos modelos, validan los resultados

---

<sup>23</sup><https://huggingface.co/xlm-roberta-base>

<sup>24</sup><https://huggingface.co/>

### **3.4 Conclusiones parciales**

---

teóricos previamente obtenidos. La manera de integrarlos como herramientas o servicios de la plataforma WisePocket, permite extender su uso a terceros.

Los resultados mostrados, asociados a la calidad de la información obtenida, y la plataforma WisePocket, contribuyen al descubrimiento de conocimiento, y a la toma de decisiones en campañas de bien público.

# CONCLUSIONES

En este apartado se resumen los resultados y aportes de mayor relevancia obtenidos en este trabajo. A lo largo de los capítulos anteriores se han expuesto con mayor detalle las conclusiones asociadas a las distintas soluciones propuestas.

En este trabajo se han presentado dos modelos computacionales: el primero de ellos para la extracción de aspectos en un flujo de opiniones relacionadas con diferentes dominios mientras que el segundo está dirigido a la clasificación de estos aspectos en el mismo contexto. Esto se logró mediante la combinación de técnicas de aprendizaje profundo y continuo, lo que permitió conservar las características y patrones comunes entre dominios.

Una de las ventajas del marco de trabajo propuesto consiste en que no necesita de la ingeniería de características en el modelo de aprendizaje profundo y continuo. Esto provocó la reducción del costo computacional, manteniendo una elevada calidad de la clasificación con respecto a otras propuestas del estado del arte. Estas técnicas fueron seleccionadas debido a que mejoran varios aspectos del procesamiento de las opiniones, por ejemplo, reducen el procesamiento de palabras que no constituyen aspectos en una oración, a la vez que reducen el olvido catastrófico que surge durante el aprendizaje incremental.

El comportamiento de las propuestas presentadas ha sido estudiado principalmente sobre colecciones de opiniones en idioma inglés, sobre varios dominios cercanos entre sí, aunque se explicaron las bases para una posible extensión al idioma español.

A lo largo de este informe han sido descritas y analizadas las principales estrategias de extracción y clasificación de aspectos que aparecen en la literatura y garantizan una elevada calidad de los resultados. De las principales técnicas estudiadas fueron seleccionadas, y comparadas con las propuestas en este informe, aquellas cuyo objetivo era el trabajo con al menos dos dominios distintos, pero que fallaron o no consiguieron una reducción apreciable del problema del olvido catastrófico, al ser aplicados a un conjunto mayor de dominios, a diferencia de lo que ocurre con las técnicas propuestas que logran manejar con mayor éxito este problema. Un resultado novedoso logrado en este trabajo es la adaptación, por primera vez, para el ABSA del algoritmo de Inteligencia Sináptica.

## Conclusiones

---

Una excepción a la situación anterior lo constituyó un trabajo de reciente aparición en la literatura, pero al ser comparado en igualdad de condiciones con las propuestas de este marco de investigación, sus resultados fueron similares o inferiores, a la vez que se demostraron sus limitaciones para ser aplicado en problemas de mayor generalidad.

Como demostración de la utilidad y validez de los modelos computacionales presentados, estos fueron aplicados en el desarrollo de una plataforma digital para campañas de bien público, donde los resultados mostraron que esta herramienta contribuye al descubrimiento de conocimiento relevante y a la toma de decisiones por parte de sus organizadores.

En definitiva, los modelos propuestos en el marco de este trabajo demuestran que es posible mantener un elevado valor de calidad en la clasificación de opiniones textales sobre conjuntos de dominios cercanos entre sí, sin que el proceso se vea afectado apreciablemente por el olvido catastrófico.

Derivadas del estudio realizado, así como de las conclusiones generales emanadas del mismo, se recomienda:

1. Estudiar la posibilidad de crear una medida de desempeño común para la estimación del olvido catastrófico.
2. Estudiar la posibilidad de combinar los métodos de regularización del cambio de los pesos en el proceso de aprendizaje de modelos de aprendizaje continuo, con propuestas de adaptación de componentes de modelos pre-entrenados como BERT.
3. Extender el estudio a propuestas de modelos para el idioma español.
4. Estudiar el comportamiento del modelo de clasificación de aspectos, con el apoyo de reglas lingüísticas para analizar el desempeño de la calidad en la clasificación.

# REFERENCIAS

- [1] K. W. Trisna and H. J. Jie, “Deep learning approach for aspect-based sentiment classification: a comparative review,” *Applied Artificial Intelligence*, pp. 1–37, 2022, ISBN: 0883-9514 Publisher: Taylor & Francis.
- [2] Z. Ke, B. Liu, H. Xu, and L. Shu, “CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6871–6883.
- [3] A. Mateen, M. Yasir, S. A. Qamar Nawaz, Q. Yasin, and M. Yunusi, “An analysis on text mining techniques for smart literature review,” *International Journal*, vol. 10, no. 2, 2021.
- [4] B. Liu, *Sentiment analysis: mining opinions, sentiments, and emotions*. Cambridge university press, 2020, pp. 116–120.
- [5] D. R. López and L. G. Arco, “Aprendizaje profundo para la extracción de aspectos en opiniones textuales,” *Revista Cubana de Ciencias Informáticas*, vol. 13, no. 2, pp. 105–145, 2019.
- [6] R. Wilson, “Cambridge analytica, facebook, and influence operations: A case study and anticipatory ethical analysis,” in *European conference on cyber warfare and security*. Academic Conferences International Limited, 2019.
- [7] L. Wang and A. P. Kirilenko, “Do tourists from different countries interpret travel experience with the same feeling? sentiment analysis of TripAdvisor reviews,” in *Information and Communication Technologies in Tourism 2021*. Springer, 2021, pp. 294–301.
- [8] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, “Deep Learning for aspect-level sentiment classification: survey, vision and challenges,” *IEEE Access*, 2019.
- [9] P. Singh and A. Manure, “Natural Language Processing with TensorFlow 2.0,” in *Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning Models with Python*. Berkeley, CA: Apress, 2020, pp. 107–129.

---

## REFERENCIAS

- [10] G. Brauwiers and F. Frasincar, “A survey on aspect-based sentiment classification,” *ACM Computing Surveys (CSUR)*, 2021, ISBN: 0360-0300 Publisher: ACM New York, NY.
- [11] L. M. Rojas-Barahona, “Deep learning for sentiment analysis,” *Language and Linguistics Compass*, vol. 10, no. 12, pp. 701–719, 2016.
- [12] E. E. Ibeke, “Computational models for contrastive opinion mining and aspect extraction,” Ph.D. dissertation, University of Aberdeen, 2018.
- [13] S. M. Al-Ghuribi, S. A. M. Noah, and S. Tiun, “Unsupervised semantic approach of aspect-based sentiment analysis for large-scale user reviews,” *IEEE Access*, vol. 8, pp. 218 592–218 613, 2020, ISBN: 2169-3536 Publisher: IEEE.
- [14] S. Wu, Y. Xu, F. Wu, Z. Yuan, Y. Huang, and X. Li, “Aspect-based sentiment analysis via fusing multiple sources of textual knowledge,” *Knowledge-Based Systems*, vol. 183, p. 104868, 2019, ISBN: 0950-7051 Publisher: Elsevier.
- [15] E. Martínez Cámara, “Análisis de Opiniones en Español,” Ph.D. dissertation, Departamento de informática, Escuela Politécnica Superior de Jaén, Mar. 2016.
- [16] Z. Chen, N. Ma, and B. Liu, “Lifelong learning for sentiment classification,” in *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2, 2015, pp. 750–756.
- [17] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [18] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, “Deep learning for aspect-based sentiment analysis: a comparative review,” *Expert Systems with Applications*, vol. 118, pp. 272–299, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417418306456>
- [19] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, pp. 228–373.
- [20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Gated feedback recurrent neural networks,” in *International Conference on Machine Learning*, 2015, pp. 2067–2075.

---

## REFERENCIAS

- [21] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, “An unsupervised neural attention model for aspect extraction,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, Vancouver, Canada, 2017, pp. 388–397.
- [22] M. Huang, Y. Wang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, USA: Association for Computational Linguistics, 2016, pp. 606–615.
- [23] M. Zhang and T. Qian, “Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 3540–3549.
- [24] D. Rothman, *Transformers for natural language processing: build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more.* Packt Publishing: Birmingham, UK, 2021, pp. 43–74.
- [25] M. Biesialska, K. Biesialska, and M. R. Costa-jussà, “Continual lifelong learning in natural language processing: a survey,” in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 6523–6541.
- [26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.
- [27] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.
- [28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [29] R. Alec, N. Karthik, S. Tim, and S. Ilya, “Improving language understanding with unsupervised learning,” Tech. Rep., Technical report, OpenAI, Tech. Rep., 2018.

---

## REFERENCIAS

- [30] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of NAACL-HLT*, 2018, pp. 2227–2237.
- [31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [32] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, “Attentional encoder network for targeted sentiment classification,” *arXiv preprint arXiv:1902.09314*, 2019.
- [33] A. Nazir, Y. Rao, L. Wu, and L. Sun, “Issues and challenges of aspect-based sentiment analysis: a comprehensive survey,” *IEEE Transactions on Affective Computing*, 2020.
- [34] X. Gu, Y. Gu, and H. Wu, “Cascaded convolutional neural networks for aspect-based opinion summary,” *Neural Processing Letters*, vol. 46, pp. 581–594, 2017.
- [35] S. Poria, E. Cambria, and A. Gelbukh, “Aspect extraction for opinion mining with a deep convolutional neural network,” *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.
- [36] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, ISBN: 0162-8828 Publisher: IEEE.
- [37] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” *Psychology of learning and motivation*, vol. 24, pp. 109–165, 1989.
- [38] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: a review,” *Neural Networks*, 2019.
- [39] V. Lomonaco, “Continual learning with deep architectures,” Ph.D. dissertation, Universidad de Bologna, Italia, 2019.
- [40] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

---

## REFERENCIAS

- [41] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabska-Barwinska, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.
- [42] L. Deng and D. Yu, “Deep learning: methods and applications,” *Foundations and Trends® in Signal Processing*, vol. 7, no. 3-4, pp. 197–387, 2014.
- [43] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: a survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, p. 1253, 2018.
- [44] H. H. Do, P. W. C. Prasad, A. Maag, and A. Alsadoon, “Deep learning for aspect-based sentiment analysis: a comparative review,” *Expert systems with applications*, vol. 118, pp. 272–299, 2019.
- [45] R. Kaur and S. Kautish, “Multimodal sentiment analysis: a survey and comparison,” *Research Anthology on Implementing Sentiment Analysis Across Multiple Disciplines*, pp. 1846–1870, 2022, publisher: IGI Global.
- [46] K. Chowdhary, “Natural language processing,” *Fundamentals of artificial intelligence*, pp. 603–649, 2020, publisher: Springer.
- [47] J. Wang, B. Xu, and Y. Zu, “Deep learning for aspect-based sentiment analysis,” in *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*. IEEE, 2021, pp. 267–271.
- [48] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, “Aspect-based sentiment analysis: a survey of deep learning methods,” *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1358–1375, 2020, ISBN: 2329-924X Publisher: IEEE.
- [49] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, and O. De Clercq, “Semeval-2016 task 5: Aspect based sentiment analysis,” in *International workshop on semantic evaluation*, 2016, pp. 19–30.
- [50] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, “SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings,” in *Thirty-Second*

## REFERENCIAS

---

- AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana USA, 2018, pp. 1795–1802.
- [51] J. D. Kelleher, B. Mac Namee, and A. D’arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020, pp. 381–594.
- [52] Z. Hai, K. Chang, and J.-j. Kim, “Implicit feature identification via co-occurrence association rule mining,” in *Computational Linguistics and Intelligent Text Processing*. Springer, 2011, pp. 393–404.
- [53] Y. Choi, E. Breck, and C. Cardie, “Joint extraction of entities and relations for opinion recognition,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 431–439.
- [54] B. Yang and C. Cardie, “Extracting opinion expressions with semi-markov conditional random fields,” in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1335–1345.
- [55] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999, pp. 50–57.
- [56] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [57] P. Lin and X. Luo, “A survey of sentiment analysis based on machine learning,” in *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 2020, pp. 372–387.
- [58] D. B. Talpur and G. Huang, “Implicit aspect based sentiment analysis for customer reviews,” *Solid State Technology*, vol. 63, no. 6, pp. 2505–2520, 2020.
- [59] D. Tang, B. Qin, and T. Liu, “Deep learning for sentiment analysis: successful approaches and future challenges,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 6, pp. 292–303, 2015.

---

## REFERENCIAS

- [60] C. Zong, R. Xia, and J. Zhang, *Text Data Mining*. Springer, 2021, vol. 711, pp. 1–14.
- [61] D. Rao and B. McMahan, *Natural language processing with PyTorch: build intelligent language applications using deep learning*. “O’Reilly Media, Inc.”, 2019, pp. 10–15.
- [62] L. Shu, B. Liu, H. Xu, and A. Kim, “Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016, pp. 225–235.
- [63] L. Shu, H. Xu, and B. Liu, “Lifelong learning crf for supervised aspect extraction,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 2, 2017, pp. 148–154.
- [64] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*. Springer, 2010, pp. 177–186.
- [65] D. Williams and G. Hinton, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–538, 1986.
- [66] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Aistats*, vol. 15, 2011, p. 275.
- [67] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [68] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, “On the importance of initialization and momentum in deep learning,” *International Conference on Machine Learning, ICML (3)*, vol. 28, no. 1139-1147, p. 5, 2013.
- [69] L. Xu, J. Lin, L. Wang, C. Yin, and J. Wang, “Deep convolutional neural network based approach for aspect-based sentiment analysis,” *Advanced Science and Technology Letters*, vol. 143, pp. 199–204, 2017.
- [70] D. Ying, J. Yu, and J. Jiang, “Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction,” in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, USA, 2017, pp. 3436–3442.

---

## REFERENCIAS

- [71] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang, “Aspect-level sentiment classification with heat (hierarchical attention) network,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. Singapore, Singapore: ACM, 2017, pp. 97–106.
- [72] P. Chen, Z. Sun, L. Bing, and W. Yang, “Recurrent attention network on memory for aspect sentiment analysis,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 452–461.
- [73] B. Huang, Y. Ou, and K. M. Carley, “Aspect level sentiment classification with attention-over-attention neural networks,” in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 2018, pp. 197–206.
- [74] L. Mai and B. Le, “Aspect-based sentiment analysis of vietnamese texts with deep learning,” in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2018, pp. 149–158.
- [75] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, “Recursive neural conditional random fields for aspect-based sentiment analysis,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, Austin, Texas, USA, 2016, pp. 616–626.
- [76] Z. Liu, Y. Lin, and M. Sun, “Representation Learning and NLP,” in *Representation Learning for Natural Language Processing*. Singapore: Springer Singapore, 2020, pp. 1–11.
- [77] M. Hoang and A. Bihorac, “Aspect-based sentiment analysis using the pre-trained language model BERT,” Ph.D. dissertation, Department of Computer Science and Engineering, Chalmers University of Technology, 2019.
- [78] I. La Vie, “Taming the hashtag: universal sentiment, SPEQ-ing the truth, and structured opinion in social media,” Ph.D. dissertation, Iowa State University, 2015.
- [79] C. Ejieh, “Aspect-based opinion mining of product reviews in microblogs using most relevant frequent clusters of terms,” Ph.D. dissertation, University of Windsor, 2016.

## REFERENCIAS

---

- [80] S. Dugar, “Aspect-based sentiment analysis using deep neural networks and transfer learning,” Ph.D. dissertation, Munich Technical University, Munich, Mar. 2019.
- [81] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [83] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, “Adapt or get left behind: domain adaptation through bert language model finetuning for aspect-target sentiment classification,” *arXiv preprint arXiv:1908.11860*, 2019.
- [84] H. Yang, B. Zeng, J. Yang, Y. Song, and R. Xu, “A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction,” *arXiv preprint arXiv:1912.07976*, 2019.
- [85] H. Ye, Z. Yan, Z. Luo, and W. Chao, “Dependency-tree based convolutional neural networks for aspect term extraction,” in *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 10235, Jeju, South Korea, 2017, pp. 350–362.
- [86] F. Xianghua, L. Guo, G. Yanyan, and W. Zhiqiang, “Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon,” *Knowledge-Based Systems*, vol. 37, pp. 186–195, 2013.
- [87] L. Deng, “A tutorial survey of architectures, algorithms, and applications for deep learning,” *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.
- [88] S. Joty, P. Liu, and H. M. Meng, “Fine-grained opinion mining with recurrent neural networks and word embeddings,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisboa, Portugal, 2015, pp. 1433–1443.
- [89] A. P. Kirilenko, S. O. Stepchenkova, H. Kim, and X. Li, “Automated sentiment analysis in tourism: comparison of approaches,” *Journal of Travel Research*, p. 0047287517729757, 2017.

## REFERENCIAS

---

- [90] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [91] S. Xiong, Y. Zhang, D. Ji, and Y. Lou, “Distance metric learning for aspect phrase grouping,” in *Proceedings of the 2016 International Conference on Computational Linguistics (Coling)*, Osaka, Japon, Dec. 2016, pp. 2492–2502.
- [92] H. Wu, Y. Gu, S. Sun, and X. Gu, “Aspect-based opinion summarization with convolutional neural networks,” in *International Joint Conference on Neural Networks (IJCNN), 2016*. IEEE, 2016, pp. 3157–3163.
- [93] L. Wang, K. Liu, Z. Cao, J. Zhao, and G. d. Melo, “Sentiment-aspect extraction based on restricted boltzmann machines,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 616–625.
- [94] B.-D. Nguyen-Hoang, Q.-V. Ha, and M.-Q. Nghiem, “Aspect-based sentiment analysis using word embeddings restricted boltzmann machines,” in *Proceedings of International Conference on Computational Social Networks*, vol. 9795. Springer, Cham, 2016, pp. 285–297.
- [95] W. Wang, V. W. Zheng, H. Yu, and C. Miao, “A survey of zero-shot learning: settings, methods, and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 13, 2019.
- [96] S. Wang, G. Lv, S. Mazumder, G. Fei, and B. Liu, “Lifelong learning memory networks for aspect sentiment classification,” in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 861–870.
- [97] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.
- [98] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–34, 2020, ISBN: 0360-0300 Publisher: ACM New York, NY, USA.

---

## REFERENCIAS

- [99] B. Liang, X. Li, L. Gui, Y. Fu, Y. He, M. Yang, and R. Xu, “Few-shot aspect category sentiment analysis via meta-learning,” *ACM Transactions on Information Systems (TOIS)*, 2022, ISBN: 1046-8188 Publisher: ACM New York, NY.
- [100] M. Gori, G. Monfardini, and F. Scarselli, “A new model for learning in graph domains,” in *Proceedings. 2005 IEEE international joint conference on neural networks*, vol. 2, 2005, pp. 729–734, issue: 2005.
- [101] H. T. Phan, N. T. Nguyen, and D. Hwang, “Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis,” *Information Sciences*, vol. 589, pp. 416–439, 2022, ISBN: 0020-0255 Publisher: Elsevier.
- [102] B. Liang, H. Su, L. Gui, E. Cambria, and R. Xu, “Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks,” *Knowledge-Based Systems*, vol. 235, p. 107643, 2022, ISBN: 0950-7051 Publisher: Elsevier.
- [103] H. Wu, Z. Zhang, S. Shi, Q. Wu, and H. Song, “Phrase dependency relational graph attention network for aspect-based sentiment analysis,” *Knowledge-Based Systems*, vol. 236, p. 107736, 2022, ISBN: 0950-7051 Publisher: Elsevier.
- [104] L. Xiao, X. Hu, Y. Chen, Y. Xue, D. Gu, B. Chen, and T. Zhang, “Targeted sentiment classification based on attentional encoding and graph convolutional networks,” *Applied Sciences*, vol. 10, no. 3, p. 957, 2020, publisher: Multidisciplinary Digital Publishing Institute.
- [105] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan, “Measuring catastrophic forgetting in neural networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana USA, 2018, pp. 3390–3398.
- [106] V. Lomonaco, L. Pellegrini, A. Cossu, A. Carta, G. Graffieti, T. L. Hayes, M. De Lange, M. Masana, J. Pomponi, and G. M. Van de Ven, “Avalanche: an end-to-end library for continual learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3600–3610.
- [107] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

---

## REFERENCIAS

- [108] F. Zenke, W. Gerstner, and S. Ganguli, “The temporal paradox of Hebbian learning and homeostatic plasticity,” *Current opinion in neurobiology*, vol. 43, pp. 166–176, 2017.
- [109] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, “Dark experience for general continual learning: a strong, simple baseline,” *Advances in neural information processing systems*, vol. 33, pp. 15 920–15 930, 2020.
- [110] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, “Memory aware synapses: Learning what (not) to forget,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.
- [111] D. Kumaran, D. Hassabis, and J. L. McClelland, “What learning systems do intelligent agents need? Complementary learning systems theory updated,” *Trends in cognitive sciences*, vol. 20, no. 7, pp. 512–534, 2016.
- [112] X. Wang, Y. Chen, and W. Zhu, “A survey on curriculum learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, ISBN: 0162-8828 Publisher: IEEE.
- [113] G. M. Van de Ven, H. T. Siegelmann, and A. S. Tolias, “Brain-inspired replay for continual learning with artificial neural networks,” *Nature communications*, vol. 11, no. 1, pp. 1–14, 2020, ISBN: 2041-1723 Publisher: Nature Publishing Group.
- [114] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, “Class-incremental learning: survey and performance evaluation on image classification,” *arXiv preprint arXiv:2010.15277*, 2020.
- [115] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, and P. Yu, “Generalizing to unseen domains: A survey on domain generalization,” *IEEE Transactions on Knowledge and Data Engineering*, 2022, ISBN: 1041-4347 Publisher: IEEE.
- [116] J. Xu and Z. Zhu, “Reinforced continual learning,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [117] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, “Progressive neural networks,” *arXiv preprint arXiv:1606.04671*, 2016.

---

## REFERENCIAS

- [118] A. Mallya and S. Lazebnik, “Packnet: Adding multiple tasks to a single network by iterative pruning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.
- [119] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, “Overcoming catastrophic forgetting with hard attention to the task,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4548–4557.
- [120] D. Lopez-Paz, “Gradient episodic memory for continual learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 6467–6476.
- [121] Z. Ke, B. Liu, N. Ma, H. Xu, and L. Shu, “Achieving forgetting prevention and knowledge transfer in continual learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 22\,443–22\,456, 2021.
- [122] A. Cossu, A. Carta, V. Lomonaco, and D. Bacciu, “Continual learning for recurrent neural networks: an empirical evaluation,” *Neural Networks*, vol. 143, pp. 607–627, 2021, ISBN: 0893-6080 Publisher: Elsevier.
- [123] J. Kim, H. Seo, W. Choi, and K. Jung, “Homeostasis-inspired continual learning: Learning to control structural regularization,” *IEEE Access*, vol. 9, pp. 9690–9698, 2021.
- [124] N. Li, C.-Y. Chow, and J.-D. Zhang, “SEML: A semi-supervised multi-task learning framework for aspect-based sentiment analysis,” *IEEE Access*, vol. 8, pp. 189\,287–189\,297, 2020, ISBN: 2169-3536 Publisher: IEEE.
- [125] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [126] D. López and L. Arco, “Multi-domain aspect extraction based on deep and lifelong learning,” in *Iberoamerican Congress on Pattern Recognition*. Springer, 2019, pp. 556–565.
- [127] F. Elsner and B. D. Wandelt, “Fast calculation of the fisher matrix for cosmic microwave background experiments,” *Astronomy & Astrophysics*, vol. 540, p. L6, 2012.
- [128] A. Madasu and A. R. Vijjini, “Sequential domain adaptation through elastic weight consolidation for sentiment analysis,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 4879–4886.

---

## REFERENCIAS

- [129] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” *arXiv preprint arXiv:1703.04200*, 2017.
- [130] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [131] Z. Ke, B. Liu, H. Wang, and L. Shu, “Continual learning with knowledge transfer for sentiment classification,” in *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, vol. 3, 2020, pp. 683–698.
- [132] G. Lv, S. Wang, B. Liu, E. Chen, and K. Zhang, “Sentiment classification by leveraging the shared knowledge from a sequence of domains,” in *International Conference on Database Systems for Advanced Applications*. Springer, 2019, pp. 795–811.
- [133] S. Poria, E. Cambria, and A. Gelbukh, “Aspect extraction for opinion mining with a deep convolutional neural network,” *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.
- [134] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.
- [135] P. Zhao, L. Hou, and O. Wu, “Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification,” *Knowledge-Based Systems*, p. 105443, 2019.
- [136] V. Lomonaco and D. Maltoni, “Core50: a new dataset and benchmark for continuous object recognition,” *arXiv preprint arXiv:1705.03550*, 2017.
- [137] F. S. Richards, “A method of maximum-likelihood estimation,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 23, no. 2, pp. 469–475, 1961, ISBN: 0035-9246 Publisher: Wiley Online Library.
- [138] C. Sutton and A. McCallum, “An introduction to conditional random fields,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012, ISBN: 1935-8237 Publisher: Now Publishers, Inc.

## REFERENCIAS

---

- [139] E. Kijsipongse and A. Piyatumrong, “A hybrid GPU cluster and volunteer computing platform for scalable deep learning,” *The Journal of Supercomputing*, vol. 74, no. 7, pp. 3236–3263, 2018, ISBN: 1573-0484 Publisher: Springer.
- [140] Z. Chen and B. Liu, “Topic modeling using topics from many domains, lifelong learning and big data,” in *International Conference on Machine Learning*, 2014, pp. 703–711.
- [141] B. Dalila, A. Mohamed, and H. Bendjanna, “A review of recent aspect extraction techniques for opinion mining systems,” in *Natural Language and Speech Processing (ICNLSP), 2018 2nd International Conference on*. IEEE, 2018, pp. 1–6.
- [142] X. Li, L. Bing, W. Zhang, and W. Lam, “Exploiting BERT for end-to-end aspect-based sentiment analysis,” in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, 2019, pp. 34–41.
- [143] F. Tang, L. Fu, B. Yao, and W. Xu, “Aspect based fine-grained sentiment analysis for online reviews,” *Information Sciences*, vol. 488, pp. 190–204, 2019.
- [144] D. Maltoni and V. Lomonaco, “Continuous learning in single-incremental-task scenarios,” *Neural Networks*, vol. 116, pp. 56–73, Aug. 2019.
- [145] K. M. Zorn, D. H. Foil, T. R. Lane, D. P. Russo, W. Hillwalker, D. J. Feifarek, F. Jones, W. D. Klaren, A. M. Brinkman, and S. Ekins, “Machine learning models for estrogen receptor bioactivity and endocrine disruption prediction,” *Environmental Science & Technology*, vol. 54, no. 19, pp. 12 202–12 213, 2020.
- [146] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, “Riemannian walk for incremental learning: Understanding forgetting and intransigence,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–547.
- [147] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, “Improving opinion aspect extraction using semantic similarity and aspect associations.” in *The Thirtieth Conference on Artificial Intelligence (AAAI-16)*. AAAI Press, 2016, pp. 2986–2992.
- [148] Z. Li, Z. G. Tian, J. W. Wang, and W. M. Wang, “Extraction of affective responses from customer reviews: an opinion mining and machine learning approach,” *International Journal of Computer Integrated Manufacturing*, pp. 1–16, 2019.

---

## REFERENCIAS

- [149] R. Singh and S. Singh, “Text similarity measures in news articles by vector space model using nlp,” *Journal of The Institution of Engineers (India): Series B*, vol. 102, no. 2, pp. 329–338, 2021.
- [150] E. Terra, A. Mohammed, and H. Hefny, “An approach for textual based clustering using word embedding,” in *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*. Springer, 2021, pp. 261–280.
- [151] B. Zeng, H. Yang, R. Xu, W. Zhou, and X. Han, “LCF: a local context focus mechanism for aspect-based sentiment classification,” *Applied Sciences*, vol. 9, no. 16, p. 3389, 2019.
- [152] R. Meyes, M. Lu, C. W. de Puiseau, and T. Meisen, “Ablation studies to uncover structure of learned representations in artificial neural networks,” in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. he Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2019, pp. 185–191.
- [153] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 1597–1607.
- [154] F. Elsner and B. D. Wandelt, “Fast calculation of the fisher matrix for cosmic microwave background experiments,” *Astronomy & Astrophysics*, vol. 540, p. L6, 2012, ISBN: 0004-6361 Publisher: EDP Sciences.
- [155] M. Hosseini, M. Horton, H. Paneliya, U. Kallakuri, H. Homayoun, and T. Mohsenin, “On the complexity reduction of dense layers from  $O(n^2)$  to  $O(n\log n)$  with cyclic sparsely connected layers,” in *2019 56th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2019, pp. 1–6.
- [156] A. Cano and B. Krawczyk, “Kappa updated ensemble for drifting data stream mining,” *Machine Learning*, vol. 109, no. 1, pp. 175–218, 2020.
- [157] F. Wu, X.-Y. Jing, Z. Wu, Y. Ji, X. Dong, X. Luo, Q. Huang, and R. Wang, “Modality-specific and shared generative adversarial network for cross-modal retrieval,” *Pattern Recognition*, vol. 104, p. 107335, 2020.

## **REFERENCIAS**

---

- [158] A. Borawska, “The role of public awareness campaigns in sustainable development,” *Economic and Environmental Studies*, vol. 17, no. 4 (44), pp. 865–877, 2017.
- [159] A. Borawska, M. Borawski, and L. Małgorzata, “The concept of virtual reality system to study the media message effectiveness of social campaigns,” *Procedia Computer Science*, vol. 126, pp. 1616–1626, 2018.
- [160] H. Kemshall and H. M. Moulden, “Communicating about child sexual abuse with the public: Learning the lessons from public awareness campaigns,” *Journal of sexual aggression*, vol. 23, no. 2, pp. 124–138, 2017.
- [161] S. Swaminathan, I. Medhi Thies, D. Mehta, E. Cutrell, A. Sharma, and W. Thies, “Learn2earn: Using Mobile Airtime Incentives to Bolster Public Awareness Campaigns,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–20, 2019.
- [162] D. L. Ramos, E. A. González, S. M. Labrada, and F. D. OFarril, “Wispocket: Plataforma digital para campañas de bien público,” in *IX Taller Internacional de Tecnologías de Software Libre y Código Abierto (Informatica 2020)*, La Habana, Mar. 2020.
- [163] M. Babst, T. Roux, and J. de Jager, “Measuring the effectiveness of out-of-home advertising campaigns in South Africa,” *Communicare: Journal for Communication Sciences in Southern Africa*, vol. 39, no. 1, pp. 33–55, 2020.
- [164] B. Park, K. Greene, and M. Colaresi, “How to teach machines to read human rights reports and identify judgments at scale,” *Journal of Human Rights*, vol. 19, no. 1, pp. 99–116, 2020.
- [165] D. K. Tayal and S. K. Yadav, “Sentiment analysis on social campaign Swachh Bharat Abhiyan using unigram method,” *AI & SOCIETY*, vol. 32, no. 4, pp. 633–645, 2017.
- [166] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, “A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate,” in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 338–343.
- [167] R. Chopade and V. Pachghare, “MongoDB Indexing for Performance Improvement,” in *ICT Systems and Sustainability*. Springer, 2020, pp. 529–539.

---

## REFERENCIAS

- [168] M. Grinberg, *Flask web development: developing web applications with python.* “O'Reilly Media, Inc.”, 2018, pp. 7–23.
- [169] T. N. Prabhu, S. Aarthi, and P. Nanthini, “Tourist sentiment analysis using natural language processing,” in *Disruptive Technologies for Big Data and Cloud Applications*. Springer, 2022, pp. 249–258.
- [170] C. Babirye, J. Nakatumba-Nabende, A. Katumba, R. Ogwang, J. T. Francis, J. Mukibi, M. Ssentanda, L. D. Wanzare, and D. David, “Building text and speech datasets for low resourced languages: a case of languages in east africa,” in *3rd Workshop on African Natural Language Processing*, 2022.
- [171] G. Qiu, B. Liu, J. Bu, and C. Chen, “Expanding domain sentiment lexicon through double propagation.” in *IJCAI*, vol. 9. Citeseer, 2009, pp. 1199–1204.
- [172] O. Day and T. M. Khoshgoftaar, “A survey on heterogeneous transfer learning,” *Journal of Big Data*, vol. 4, no. 1, p. 29, 2017.
- [173] X. Fang and J. Tao, “A transfer learning based approach for aspect based sentiment analysis,” in *2019 sixth international conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 478–483.
- [174] Y. Duan, M. Andrychowicz, B. Stadie, O. J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, “One-shot imitation learning,” in *Advances in neural information processing systems*, 2017, pp. 1087–1098.
- [175] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, 2021, ISBN: 0925-2312 Publisher: Elsevier.
- [176] M.-F. Li, K. Zhou, H. Wang, L. Ma, and X. Li, “Aspect-based sentiment classification with reinforcement learning and local understanding,” in *International Conference on Artificial Neural Networks*. Springer, 2021, pp. 662–674.
- [177] V. A. Rao, K. Anuranjana, and R. Mamidi, “A sentiwordnet strategy for curriculum learning in sentiment analysis,” in *International Conference on Applications of Natural Language to Information Systems*. Springer, 2020, pp. 170–178.

## **REFERENCIAS**

---

- [178] A. Rogers, O. Kovaleva, and A. Rumshisky, “A primer in bertology: What we know about how bert works,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020.

# **ANEXO A**

## **Producción científica del autor**

### **A.1. Publicaciones en revistas y conferencias**

1. López, D. y Arco, L.[2018]. “Aprendizaje profundo para la extracción de aspectos: tarea esencial en la creación y uso de las ontologías”, *Proceedings of the 3rd International Workshop on Semantic Web (IWSW)*, pp. 143-156. **Indexado en CEUR-WS**.
2. López, D. y Arco, L.[2019]. “Aprendizaje profundo para la extracción de aspectos en opiniones textuales”, *Revista Cubana de Ciencias Informáticas*, vol. 13, núm. 2, pp. 105-145. **Indexado en Scielo**.
3. López, D.y Arco, L. [2019]. “Multi-domain aspect extraction based on deep and lifelong learning”, *In Iberoamerican Congress on Pattern Recognition*, Springer, Cham, pp. 556-565. **Indexado en Springer (Web of Sciences)**.
4. López, D., Astorga, E., Morejon, S. y Deller, O. [2020]. “Wispocket: Plataforma digital para campañas de bien público”, *Memorias del IX Taller Internacional de Tecnologías de Software Libre y Código Abierto (Informática 2020)*, ISBN:978-959-7255-01-7, La Habana, Cuba.
5. Gil, M., López, D. y Herold, S. [2021]. “Digital content editing system for smartphones (athrim 1.0)”, *In Cross Reality and Data Science in Engineering*, Cham, pp. 762-772. **Indexado en Springer (Web of Sciences)**.
6. López, D. y Artigas-Fuentes, F. [2022]. “Un modelo de aprendizaje continuo y profundo para la clasificación de sentimientos basado en aspectos”, *Memorias de la VI Conferencia Internacional en Ciencias Computacionales e Informáticas (CICCI 2022)*, ISBN: 978-959-7255-02-4, La Habana, Cuba.

#### **A.1.1. Otras producciones científicas relacionadas con el investigador**

- Gil, M., López, D., Herold, S. y Márquez A. [2019]. “Sistema de edición de contenidos de salud para teléfonos inteligentes”, *Revista Cubana de Informática Médica*, vol. 11,

## **A.2 Participación en eventos internacionales y nacionales**

---

núm. 2, pp. 195-201. *Indexado en Scielo*.

- Mengana-de la Fe, G. y López, D. [2019]. “Realidad aumentada, una herramienta para la gestión de los valores patrimoniales”, *Revista Santiago*, núm. 149, pp. 213-222. *Indexado en DOAJ*.

## **A.2. Participación en eventos internacionales y nacionales**

1. VI Conferencia Internacional en Ciencias Computacionales e Informáticas (CICCI 2022), La Habana, Cuba.
2. XIII Seminario Nacional sobre Estudios Canadienses, 2022, Santiago de Cuba, Cuba.
3. 1er Hackaton de Procesamiento de Lenguaje Natural, 2022.
4. 1er Taller Regional La Inteligencia Artificial en la Transformación Digital, 2022, Santiago de Cuba, Cuba.
5. IX Taller Internacional de Tecnologías de Software Libre y Código Abierto (Informática 2020), La Habana, Cuba.
6. International Iberoamerican Congress on Pattern Recognition, 2019, La Habana, Cuba.
7. International Conference on Interactive Collaborative and Blended Learning, 2019, Santiago de Cuba, Cuba.
8. Ganador de premio iStartUp (Gran Premio) por la creación de la plataforma WisePocket, en el evento Internacional Digital For Development (D4D), auspiciado por la Agencia de Cooperación del Reino de Bélgica 2018, Bruselas, Bélgica.
9. The 3rd International Workshop on Semantic Web 2018, La Habana, Cuba.

## **A.3. Registros de software**

1. Athrim 1.0 (Sistema de edición de contenidos digitales para teléfonos inteligentes) Registro: 0946-03-2019.
2. Plataforma Informática para la gestión de campañas de bien público, versión 1.0 Registro: 3595-10-2019.

## **ANEXO B**

### **Principales características de los modelos de aprendizaje profundo**

---

<b>Modelo</b>	<b>Tipo de aprendizaje</b>	<b>Características</b>
Autoencoders (AE)	No supervisado	<ul style="list-style-type: none"> <li>■ Adecuado para la extracción de características, reducción de dimensionalidad.</li> <li>■ El mismo número de unidades de entrada que de salida.</li> <li>■ Permite la reconstrucción de los datos de entrada en la capa de salida.</li> <li>■ Trabaja con datos no etiquetados.</li> </ul>
Redes Neuronales Recurrentes ( <i>Recurrent Neural Network</i> ; RNN)	Supervisado	<ul style="list-style-type: none"> <li>■ Problemas de series de tiempo.</li> <li>■ Procesa una secuencia de datos a través de una memoria interna.</li> <li>■ Útil en problemas donde los datos dependen del tiempo (p.ej., NLP, Procesamiento de audio)</li> </ul>
Memorias de corto plazo ( <i>Long Short Term Memory</i> ; LSTM)	Supervisado	<ul style="list-style-type: none"> <li>■ Problemas de series de tiempo.</li> <li>■ Buen desempeño con datos de tamaño variable y dependientes del tiempo (p.ej., NLP, fotogramas de video).</li> <li>■ El acceso a las celdas de memoria es protegido por compuertas.</li> </ul>

Tabla B.1: Modelos de aprendizaje profundo usados en trabajos sobre ABSA.

## **ANEXO C**

### **Diagrama del proceso de entrenamiento del modelo de extracción de aspectos**

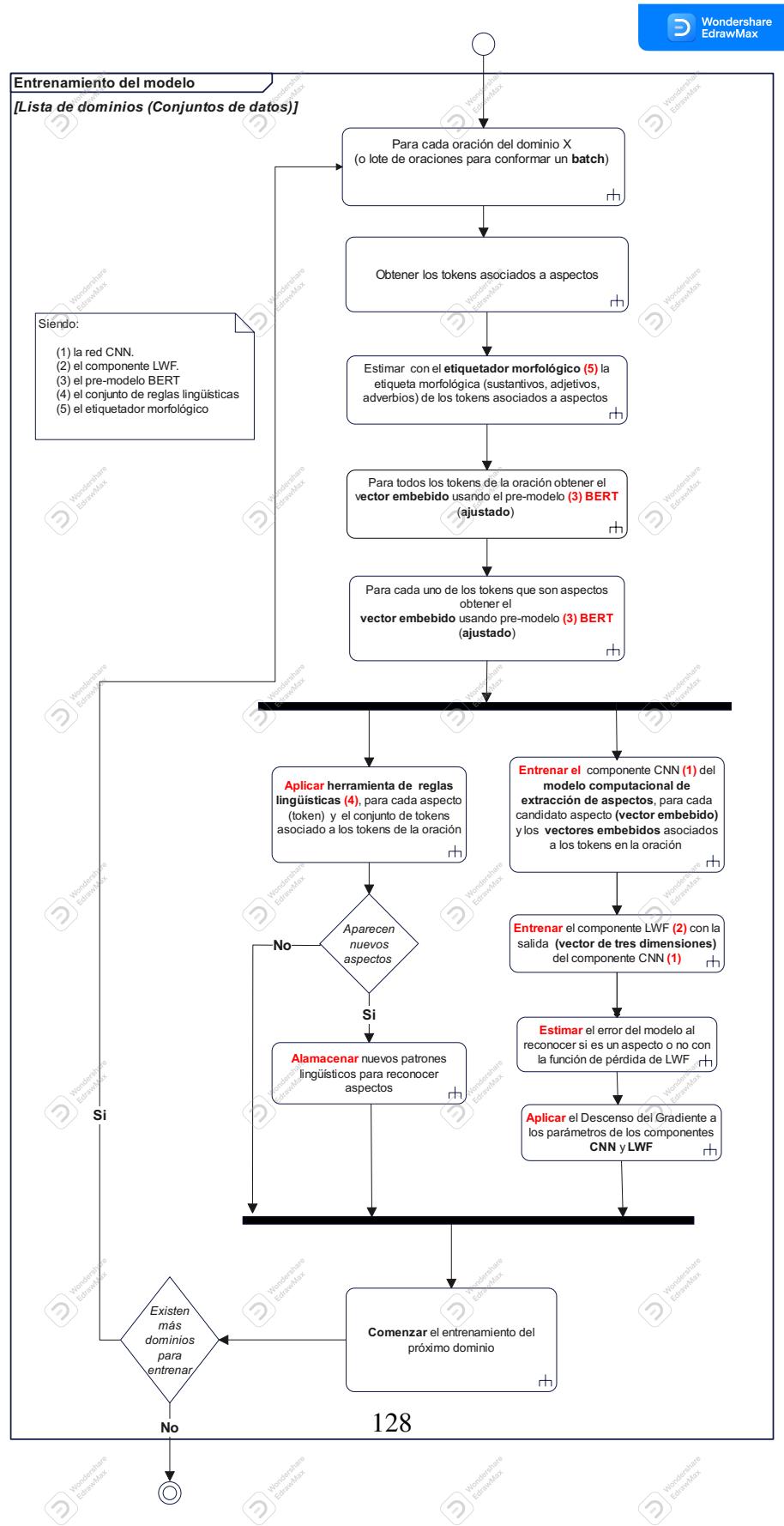


Figura C.1: Diagrama del flujo del entrenamiento del modelo Lwf-CNN-lgR.

## **ANEXO D**

### **Diferencias entre aprendizaje continuo y otras estrategias**

<b>Paradigma</b>	<b>Diferencias</b>
<b>Aprendizaje por transferencia (<i>Transfer Learning</i>)</b>	<p>Este paradigma involucra dos dominios: un dominio fuente y un dominio destino. Aunque puede existir más de una fuente, es una práctica común usar un solo dominio fuente [16, 172, 173].</p> <p>Este paradigma se diferencia del aprendizaje continuo en que la transferencia de conocimiento ocurre una sola vez (dominio fuente a dominio destino). No retiene el conocimiento transferido o información (p.ej.; pesos en las neuronas) para usos futuros [173].</p>

---

Paradigma	Diferencias
<b>Aprendizaje multitarea (<i>Multi-task learning</i>; MTL)</b>	<p>Este paradigma aprende varias tareas simultáneamente, intentando obtener un mejor desempeño al usar la información relevante en varias tareas [124]. El uso de un sesgo o bias permite explotar la relevancia de la estructura y evitar el sobreentrenamiento en cada tarea. Las que se entranan simultáneamente deben tener similitud en cuanto al dominio de aplicación (p.ej.; procesamiento de imágenes, NLP, procesamiento de audio, entre otras).</p> <p>Con respecto al aprendizaje continuo, ambos paradigmas usan la misma información compartida entre tareas para ayudar al aprendizaje. El aprendizaje multitarea, en lugar de optimizar una sola tarea, optimiza varias simultáneamente. Al no acumular ningún conocimiento en el tiempo, para ser empleado en la próxima tarea o dominio, no realiza un aprendizaje continuo [16, 39].</p>

---

Paradigma	Diferencias
<b>Meta-aprendizaje (<i>Meta-Learning</i>; ML)</b>	<p>Es conocido como <i>aprendiendo a aprender</i> [174]. Es un proceso de aprendizaje que emplea metadatos de experiencias anteriores con el objetivo de mejorar la capacidad de aprendizaje de nuevas experiencias [174]. Aunque este paradigma parece compartir algunos objetivos importantes del aprendizaje continuo. Realiza el entrenamiento de los modelos sobre un conjunto de datos fijo o con un dominio objetivo específico, sin considerar otros dominios como fuente. Una constante en este paradigma es la presencia de un sistema dual: uno para aprender en la tarea actual y el segundo para orientar el proceso de aprendizaje [16, 38, 39].</p>

---

Paradigma	Diferencias
<b>Aprendizaje en línea (<i>Online learning</i>)</b>	<p>En este paradigma los datos de entrenamiento son procesados en un momento del tiempo. Cuando el nuevo dato aparece, el modelo existente es rápidamente actualizado para producir uno más eficaz. Este paradigma se centra en proponer modelos que aprendan eficientemente cuando aparece un nuevo dato.</p> <p>En el caso del aprendizaje continuo, se realiza el aprendizaje sobre una secuencia de diferentes grupos/tareas y trata de encontrar un balance entre el aprendizaje anterior y el actual, manteniendo una base de conocimientos.</p> <p>En este proceso, el aprendizaje del nuevo dato no debe comprometer el desempeño al evaluar datos anteriores [16, 39, 175]. Este paradigma es el más cercano al aprendizaje continuo.</p>

---

Paradigma	Diferencias
<b>Aprendizaje por reforzamiento (Reinforcement Learning)</b>	Se orienta a problemas donde los agentes aprenden acciones a través del método de prueba y error interactuando con ambientes dinámicos. En cada paso de interacción, los agentes reciben como entrada el estado actual del ambiente y una posible recompensa. El objetivo es aprender una política óptima que convierte los estados en acciones y maximiza la futura recompensa esperada [176]. Los preceptos de este paradigma no están relacionados con el aprendizaje continuo.
<b>Aprendizaje curricular (Curriculum Learning)</b>	Propone una secuencia de datos o tareas a aprender por un modelo. Esto lo relaciona con el aprendizaje continuo. Sin embargo, las tareas son elegidas y estructuradas de forma que hacen posible aprender más eficientemente la última, pero teniendo en cuenta las diferencias, dificultades y dependencias funcionales entre ellas [112, 177], mientras que en el aprendizaje continuo, las tareas no son voluntariamente seleccionadas.

## **ANEXO E**

### **Módulos de la plataforma Wisepocket**

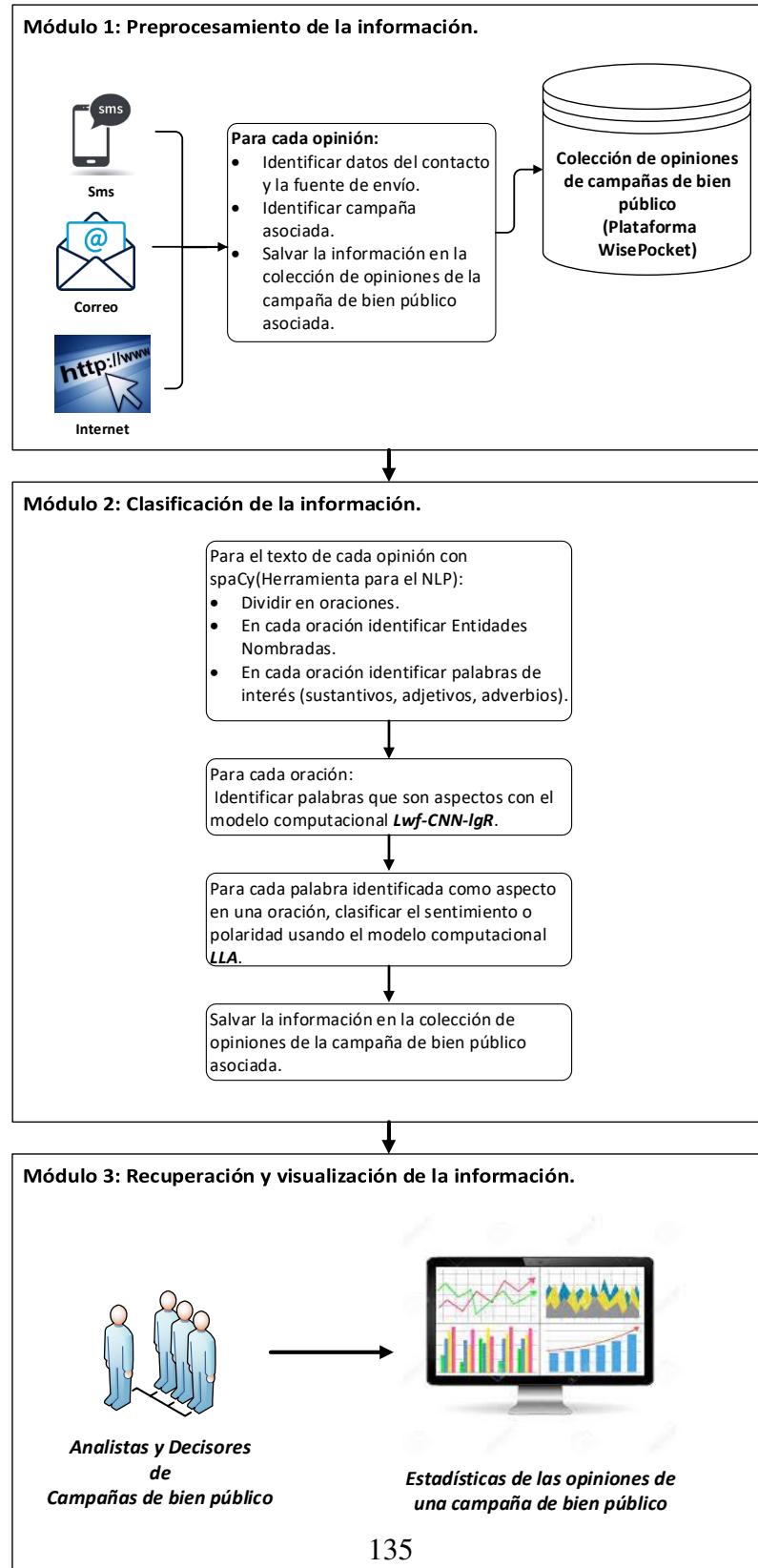


Figura E.1: Módulos de la plataforma WisePocket.

## **ANEXO F**

### **Trabajos que combinan el aprendizaje continuo y el análisis de sentimientos**

<b>Título del Artículo</b>	<b>Año</b>	<b>Tarea del NLP</b>	<b>Olvido catastrófico</b>	<b>Observaciones</b>
Continual Learning with Knowledge Transfer for Sentiment Classification	2020	Análisis de sentimientos	Una vez entrenado el último clasificador, promediar los resultados en cada conjunto de pruebas de las tareas anteriores a la última.	A partir del concepto del olvido catastrófico, un método que haya perdido todo el conocimiento anterior (valores de los pesos de sus redes neuronales) no debería mostrar altos valores en esta medida.
Cross-Domain End-To-End Aspect-Based Sentiment Analysis with Domain-Dependent Embeddings	2021	Análisis de sentimientos	Valores de F1-micro con un máximo de 0.62.	

<b>Título del Artículo</b>	<b>Año</b>	<b>Tarea del NLP</b>	<b>Olvido catastrófico</b>	<b>Observaciones</b>
Sequential Domain Adaptation through Elastic Weight Consolidation for Sentiment Analysis	2021	Análisis de sentimientos	Exactitud de 0.78 para el uso de modelo que combina LSTM y EWC y el dominio de prueba son conjuntos de datos de opiniones sobre productos electrónicos y para un modelo que relaciona un CNN y un EWC tiene 0.74 con el dominio de prueba de opiniones de reproductores de DVD.	Todos los resultados reportados se promedian sobre cinco ciclos o epoch. El resultado final es sobre un dominio objetivo (los dominios anteriores son de entrenamiento).
A Sentiwordnet Strategy for Curriculum Learning in Sentiment Analysis	2020	Análisis de sentimientos	Exactitud de 0.45.	Resultados de calidad muy bajos con respecto al estado del arte.
Sentiment Classification by Leveraging the Shared Knowledge from a Sequence of Domains	2019	Análisis de sentimientos	Exactitud de 0.89.	

Título del Artículo	Año	Tarea del NLP	Olvido catastrófico	Observaciones
Using the Past Knowledge to Improve Sentiment Classification	2020	Análisis de sentimientos	Exactitud de 0.69.	Entrenan todos los algoritmos y presentan como resultado de la exactitud del modelo en la última tarea entrenada. Este resultado muestra el promedio de la exactitud global.
DomBERT: Domain-oriented Language Model for Aspect-based Sentiment Analysis	2020	ABSA	Para el ASC se calcula la exactitud y el F1-macro sobre 3 clases de polaridad Laptop exactitud de 0.77 Laptop F1-macro de 0.73 Restaurante exactitud e 0.83 Restaurante F1-macro de 0.75.	Fue tomado porque referencia a varios artículos de aprendizaje continuo basado en el análisis de sentimientos.
Bayes-enhanced Lifelong Attention Networks for Sentiment Classification	2020	Análisis de Sentimientos	Exactitud (opiniones en Amazon) de 0.92 y F1-macro (opiniones en Amazon) de 0.75, exactitud (SNAP 24) de 0.95 y F1-macro de 0.70.	

---

<b>Título del Artículo</b>	<b>Año</b>	<b>Tarea del NLP</b>	<b>Olvido catastrófico</b>	<b>Observaciones</b>
Disentangling Aspect and Opinion Words in Sentiment Analysis using Lifelong PU Learning	2018	Análisis de sentimientos	0.84 para acc@150.	Se emplea la exactitud (acc@n) como medida de evaluación, donde n es el conjunto de 50, 100, y 150 palabras. Dado un objetivo, se coleccionan aquellas t-palabras más mencionadas, y manualmente se etiquetan estas como palabras de opinión o aspecto y luego se evalúa con respecto a lo que obtienen los métodos.
Lamol: Language modeling for lifelong language learning	2020	Varias tareas incluyendo el análisis de sentimientos	Todas las tareas usan la exactitud y presentan la desviación estándar. Mayor valor de exactitud de 0.80 con una desviación estándar de 0.8.	Esta propuesta entrena de forma secuencial diversas tareas como: preguntas y respuestas, parsers semánticos y análisis de sentimientos.

---

<b>Título del Artículo</b>	<b>Año</b>	<b>Tarea del NLP</b>	<b>Olvido catastrófico</b>	<b>Observaciones</b>
Projecting embeddings for domain adaptation: Joint modeling of sentiment analysis in diverse domains	2018	Análisis de sentimientos	Se reporta el valor de Exactitud para un conjunto de datos balanceado de opiniones de Amazon y el F1-macro para las pruebas con el conjunto de datos de SemEval. La exactitud de 0.85 y el F1-macro de 0.67.	Este trabajo está enfocado más al aprendizaje por transferencia que al aprendizaje continuo o lifelong porque los resultados se muestran por la evaluación de un dominio fuente hacia un dominio destino.
Lifelong learning for sentiment classification	2015	Análisis de sentimientos	Se usa el F1-macro por el desbalance de las clases con un valor de 0.67	No emplea modelos de aprendizaje profundo.
Lifelong Learning of Few-shot Learners across NLP Tasks.	2021	Varias tareas incluyendo el análisis de sentimientos	Promedio de la exactitud de 0.80	Se reporta el valor de la exactitud en cada tarea entrenada, después de entrenar cada tarea y después de entrenar todas las tareas.

# ANEXO G

## Representación textual

El módulo de análisis de sentimientos trabaja con textos (datos no estructurados), por tanto, la representación textual es indispensable para su procesamiento posterior por el módulo de clasificación de ABSA [8]. Para la construcción del modelo de representación, la herramienta spaCy realiza la detección de las oraciones y en ellas los *tokens* o palabras [11]. Para todas las palabras (excepto las que constituyen artículos, conjunciones u otros símbolos de parada), se usa el modelo computacional para la extracción de aspectos Lwf-CNN-lgR para determinar si es un aspecto o no [126]. Para el conjunto de aspectos detectados, se identifica su polaridad o sentimiento con el uso del modelo para la clasificación de aspectos LLA [11].

La forma de representación para el texto de las oraciones y los aspectos es similar a la del modelo BERT [31], que es la capa de entrada de los modelos de aprendizaje continuo Lwf-CNN-lgR y LLA. Para cada oración presente en el texto de una opinión, las palabras y otros elementos textuales son tratados como *tokens*.

En una oración perteneciente a una opinión, para cada *token* su representación de entrada es construida por la suma de vectores de la representación vectorial (embeddings) del *token*, el segmento y la posición. Un ejemplo de esta representación se puede ver en la figura G.1.

Para obtener la representación vectorial de un *token* se emplea el WordPiece [178]. Este contiene 30 mil *tokens* en su vocabulario. Se denota la división de conjuntos de palabras con #.

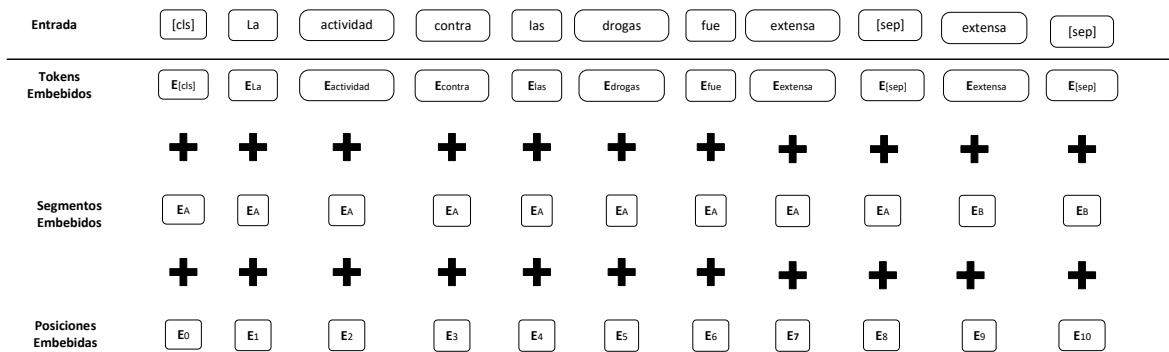


Figura G.1: Ejemplo del vector embebido que se obtiene como entrada al modelo BERT a partir de una oración.

- 
- Se usa una representación vectorial para la posición en que aparece la palabra, con longitudes de secuencia de palabras o *tokens* de hasta 512.
  - El primer *token* de una secuencia siempre es la representación especial [CLS]. La salida de un Transformer corresponde a este *token* y su función principal es como representación de una secuencia de entrada para una tarea de clasificación supervisada del PLN. En tareas donde no se realiza la clasificación este vector es ignorado.
  - Cada una de las oraciones y *tokens* candidatos a aspectos son colocados juntos en una sola secuencia, pero separados con el *token* especial [SEP].

En la figura G.1 se muestra la representación textual para la oración “*La actividad contra las drogas fue extensa.*” y el posible aspecto *extensa* (Se ilustra un ejemplo en español para una mejor compresión del texto).

Para la obtención de la representación embebida de los *tokens* de una oración, las palabras como *playing* (jugando) es dividida en palabras como "play' y "#ing'. Esto ayuda de dos formas:

- Limita el tamaño del vocabulario al no tener necesidad de usar varias formas de representación de la misma palabra (p.ej., playing, plays, players, etc.).
- Ayuda con palabras que no se encuentran en el vocabulario. Por ejemplo, si *playing* no aparece en el vocabulario se puede sustituir por los vectores embebidos de "play' y "#ing'.

Para obtener este vector asociado al *token* se toma de una matriz de tamaño 30000 x 768 (dimensión), donde, 30 mil es el tamaño del vocabulario en WordPiece [31].

El vector embebido del segmento de texto es empleado para indicar el segmento de la oración al que pertenece el aspecto. Por ejemplo, todos los *tokens* cercanos al aspecto en la oración (a partir de una distancia o radio de palabras cercanas) pueden ser representados por un vector inicializado en unos y de dimensión 768 y los que no se encuentran en el radio o vecindad del posible aspecto se representan con un vector inicializado con ceros y de dimensión 768. El *token* candidato a aspecto es representado por un vector inicializado con unos, de igual dimensión a los vectores anteriores.

Para el vector embebido de las posiciones de cada *token* en la secuencia de palabras se tiene una

---

matriz de valores constantes, donde la primera fila es la asociada al *token* [CLS], y la segunda para el resto de los *tokens* de la oración. Para determinar los valores del vector embebido de 768 dimensiones se tiene en cuenta la posición del *token* en la oración, con el uso de las funciones de seno y coseno [178]:

$$PE(pos, 2i) = \sin(pos/10000^{2i/D})$$

$$PE(pos, 2i + 1) = \cos(pos/10000^{2i/D})$$