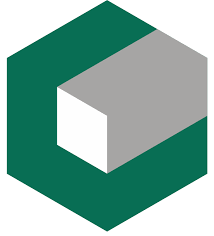
**Universidad Tecnológica de La Habana**

**“José Antonio Echeverría”**

**CUJAE**



**Facultad de Ingeniería Informática**

**Aplicación de modelos *Transformers* al Análisis de Sentimientos basado en aspectos para idioma Español**

***Trabajo de diploma presentado en opción al título de Ingeniero Informático***

**Autor**: Miguel Angel Rivero Tapia

**Tutor**: Dr. C. Alfredo Javier Simón Cuevas

**La Habana, Cuba**

**Marzo 2024**

**Resumen**

En la actualidad, el auge de la información textual disponible en línea, proveniente de fuentes como noticias, reseñas de productos y redes sociales, ha incrementado la necesidad de contar con métodos automáticos para analizarla de manera rápida y precisa. Este análisis es crucial para la toma de decisiones, lo que ha llevado a grandes esfuerzos por identificar el sentimiento en textos no estructurados. Sin embargo, en un contexto donde las opiniones se basan en aspectos específicos de un producto o servicio, es esencial ir más allá del análisis general y enfocarse en los sentimientos asociados a cada aspecto individual. El análisis de sentimientos basado en aspectos (ABSA, por sus siglas en inglés) se centra en comprender el sentimiento de los usuarios hacia características específicas de una entidad. No obstante, identificar aspectos y determinar su polaridad sigue siendo un desafío, especialmente en idiomas como el español, donde la mayoría de las soluciones existentes han sido desarrolladas y evaluadas principalmente para el inglés, lo que resulta en una menor precisión para otros idiomas. Esta tesis propone un nuevo enfoque para el ABSA utilizando ensambles de Transformers. La idea central combina la potencia de los Transformers con técnicas de ensamblado de modelos, aprovechando las ventajas de diferentes arquitecturas y estrategias de aprendizaje. El método consta de tres etapas clave: pre-procesamiento del texto, extracción de aspectos y predicción de polaridad. La solución fue evaluada utilizando el conjunto de datos de SemEval 2016 Task 5, específicamente diseñado para ABSA, demostrando una mejora significativa en la precisión y robustez del análisis para el idioma español.

**Palabras clave**: Aprendizaje profundo, Transformers, análisis de sentimientos, extracción de aspectos, ensambles, español.

**Abstract**

**Keywords:** Deep learning, Transformers, sentiment analysis, aspect extraction, ensemble.

**Índice:**

[Introducción 1](#_Toc191836224)

[Capítulo 1: Análisis de sentimientos basado en aspectos. 8](#_Toc191836225)

[1.1. Procesamiento del Lenguaje Natural (PLN). 8](#_Toc191836226)

[1.1.1. Niveles de análisis lingüístico en PLN 9](#_Toc191836227)

[1.2. Análisis de Sentimientos. 10](#_Toc191836228)

[1.2.1. Niveles del análisis de sentimientos. 11](#_Toc191836229)

[1.3. Análisis de sentimientos basado en aspectos. 12](#_Toc191836230)

[1.4. Enfoques utilizados en el ABSA 13](#_Toc191836231)

[1.4.1. Enfoque no supervisado. 13](#_Toc191836232)

[1.4.2. Enfoque supervisado. 14](#_Toc191836233)

[1.5. Trabajos relacionados. 16](#_Toc191836234)

[1.6. Redes neuronales Transformers. 18](#_Toc191836235)

[1.7. Modelos pre-entrenados de lenguaje con arquitectura Transformers. 20](#_Toc191836236)

[1.7.1. BERT (*Bidirectional Encoder Representation from Transformers*). 20](#_Toc191836237)

[1.7.2. ALBERT (*A Lite BERT*). 21](#_Toc191836238)

[1.7.3. RoBERTa (*Robustly optimized BERT approach*). 22](#_Toc191836239)

[1.7.4. GPT-2 (*Generative Pre-trained Transformer 2*). 23](#_Toc191836240)

[1.7.5. ELECTA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately). 24](#_Toc191836241)

[1.7.6. Comparación de las arquitecturas de los modelos utilizados 25](#_Toc191836242)

[1.8. Técnicas de ensamble de clasificadores. 25](#_Toc191836243)

[1.8.1. Método de votación (*Voting Ensemble*). 26](#_Toc191836244)

[1.8.1.1. Votación máxima. 26](#_Toc191836245)

[1.8.1.2. Votación promedio. 27](#_Toc191836246)

[1.8.1.3. Votación promedio ponderada. 27](#_Toc191836247)

[1.8.2. Método de metaaprendizaje. 28](#_Toc191836248)

[1.8.3. Método de embolsado (*Bagging*). 29](#_Toc191836249)

[1.8.4. Método de apilamiento (*Stacking*). 29](#_Toc191836250)

[1.8.5. Método de reforzamiento (*Boosting*). 30](#_Toc191836251)

[1.9. Métodos de evaluación. 30](#_Toc191836252)

[1.9.1. Conjunto de datos. 31](#_Toc191836253)

[1.9.2. Métricas utilizadas 32](#_Toc191836254)

[1.9.2.1. Precisión 32](#_Toc191836255)

[1.9.2.2. Recall 33](#_Toc191836256)

[1.9.2.3. F1-Score 33](#_Toc191836257)

[1.9.2.4. Accuracy 33](#_Toc191836258)

[1.10. Conclusiones parciales. 33](#_Toc191836259)

[Capítulo 2: Solución de análisis de sentimientos basado en aspectos para idioma español. 34](#_Toc191836260)

[2.1. Descripción de la solución. 35](#_Toc191836261)

[2.1.1. Preprocesamiento 36](#_Toc191836262)

[2.1.2. Extracción de aspectos. 37](#_Toc191836263)

[2.1.3. Determinación de la polaridad. 39](#_Toc191836264)

[2.2. Desarrollo de la solución de análisis de sentimientos basado en aspectos. 41](#_Toc191836265)

[2.2.1. Modelo de dominio. 41](#_Toc191836266)

[2.2.2. Glosario de términos. 42](#_Toc191836267)

[2.2.3. Captura de requisitos. 43](#_Toc191836268)

[2.2.3.1. Requisitos funcionales. 43](#_Toc191836269)

[2.2.3.2. Requisitos no funcionales. 44](#_Toc191836270)

[2.2.4. Diagrama de caso de uso del sistema. 44](#_Toc191836271)

[2.2.5. Descripción detallada de los casos de uso. 45](#_Toc191836272)

[2.2.6. Descripción de paquetes. 51](#_Toc191836273)

[2.2.7. Descripción de clases. 52](#_Toc191836274)

[2.3. Problemas frecuentes. 53](#_Toc191836275)

[2.3.1. Insuficiencia de datos. 54](#_Toc191836276)

[2.3.2. Desbalance de clases. 54](#_Toc191836277)

[2.3.3. Sobreajuste (*Overfitting*). 54](#_Toc191836278)

[2.4. Conclusiones parciales. 55](#_Toc191836279)

[Capítulo 3: Evaluación de la solución propuesta. 56](#_Toc191836280)

[3.1. Marco de Evaluación. 56](#_Toc191836281)

[3.2. Descripción de las fuentes de datos 57](#_Toc191836282)

[3.3. Resultados y discurso. 58](#_Toc191836283)

[3.3.1. Resultados de la evaluación de la extracción de aspectos. 58](#_Toc191836284)

[3.3.2. Análisis de los resultados de la extracción de aspectos. 60](#_Toc191836285)

[3.3.3. Resultados de la clasificación de la polaridad. 61](#_Toc191836286)

[3.3.4. Análisis de los resultados de la clasificación de la polaridad. 64](#_Toc191836287)

[3.4. Conclusiones parciales. 65](#_Toc191836288)

[Conclusiones generales. 66](#_Toc191836289)

[Recomendaciones. 67](#_Toc191836290)

[Referencias Bibliográficas 68](#_Toc191836291)

**Índice de tablas.**

[Tabla 1. Soluciones de ABSA reportadas en la literatura 16](#_Toc191836199)

[Tabla 2. Comparación de las arquitecturas de los modelos utilizados 25](#_Toc191836200)

[Tabla 3. Glosario de término. 42](#_Toc191836201)

[Tabla 4. Descripción del CUS “Seleccionar fuente de datos”. 45](#_Toc191836202)

[Tabla 5. Descripción del CUS “Preprocesar datos”. 46](#_Toc191836203)

[Tabla 6. Descripción del CUS “Entrenar modelos”. 46](#_Toc191836204)

[Tabla 7. Descripción del CUS “Modelar para aspectos”. 47](#_Toc191836205)

[Tabla 8. Descripción del CUS “Modelar para clasificación”. 48](#_Toc191836206)

[Tabla 9. Descripción del CUS “Predecir polaridad”. 49](#_Toc191836207)

[Tabla 10. Descripción del CUS “Extraer aspectos”. 50](#_Toc191836208)

[Tabla 11. Descripción del CUS “Clasificar opinión”. 50](#_Toc191836209)

[Tabla 12.Descripción del CUS “Visualizar resultados”. 51](#_Toc191836210)

[Tabla 13. Descripción de paquetes 52](#_Toc191836211)

[Tabla 14. Descripción de las clases utilizadas. 53](#_Toc191836212)

[Tabla 15. Caracterización del corpus ABSA-SemEval-2016 58](#_Toc191836213)

[Tabla 16. Configuración de los hiperparámetros de los modelos. 58](#_Toc191836214)

[Tabla 17. Evaluación de la calidad de la extracción de aspectos. 59](#_Toc191836215)

[Tabla 18. Evaluación del ensamble por votación máxima 60](#_Toc191836216)

[Tabla 19. Comparación de la solución para la extracción de aspectos frente a otras soluciones reportadas en la literatura. 60](#_Toc191836217)

[Tabla 20. Configuración de los hiperparámetros de los modelos 61](#_Toc191836218)

[Tabla 21. Análisis de los resultados de diferentes tamaño de ventana. 62](#_Toc191836219)

[Tabla 22. Resultados de polaridad con el método de ventana de palabras (tamaño 6) 62](#_Toc191836220)

[Tabla 23. Resultados de polaridad con el método de análisis de dependencias 63](#_Toc191836221)

[Tabla 24. Resultados de polaridad con el método hibrido 63](#_Toc191836222)

[Tabla 25. Comparación de la solución propuesta para clasificación de polaridad 64](#_Toc191836223)

**Índice de figuras.**

[Figura 1. Flujo de ensamble por votación máxima. 27](#_Toc191822128)

[Figura 2. Esquema general del flujo del ABSA utilizando un enfoque supervisado. 36](#_Toc191822129)

[Figura 3. Representación de las reseñas. 37](#_Toc191822130)

[Figura 4. Ejemplo de identificación de aspectos con el uso del método de ensamble de votación promedio. 39](#_Toc191822131)

[Figura 5. Modelo de dominio. 42](#_Toc191822132)

[Figura 6.Diagrama de casos de uso. 45](#_Toc191822133)

# Introducción

La proliferación de datos textuales en internet tales como blogs, prensa en línea, webs comparativas, evaluación de productos y servicios, especialmente en redes sociales, ha generado un interés creciente en el análisis de sentimientos. Este interés se debe a que la información que se genera en línea, a través de textos, refleja las opiniones, emociones y valoraciones de las personas sobre un sinfín de temas. Esta información se convierte en un recurso valioso para diversas áreas, como el marketing, la gestión de la reputación, la investigación de mercados y la atención al cliente.

El análisis de sentimientos, es el campo de estudio que analiza las opiniones, sentimientos, valoraciones, actitudes y emociones de las personas hacia entidades como productos, servicios, organizaciones, individuos, problemas, eventos, temas y sus atributos. En la literatura se le conoce con muchos nombres y tareas ligeramente diferentes, por ejemplo, minería de opiniones, extracción de opiniones, minería de sentimientos, análisis de subjetividad, análisis de afectos, análisis de emociones, minería de reseñas, entre otras [[1](#_ENREF_1), [2](#_ENREF_2)]

Su relevancia se extiende a diversas áreas, como el análisis de reseñas de productos, la monitorización de la opinión pública o la predicción de las tendencias de consumo. El análisis de sentimientos proporciona a empresas y organizaciones una herramienta poderosa para comprender mejor la percepción del público y tomar decisiones estratégicas más acertadas[[2](#_ENREF_2), [3](#_ENREF_3)].

El análisis de sentimientos ha sido objeto de estudio desde diversas perspectivas, centrándose principalmente en tres niveles: documento, oración y aspecto. Cada uno de estos niveles ofrece un grado de detalle diferente en el análisis de las opiniones expresadas en los textos, permitiendo una comprensión más profunda de la información emocional que subyace en el lenguaje [[2](#_ENREF_2), [4](#_ENREF_4)].

En el nivel de documento, el objetivo principal es determinar la polaridad general de un texto. Se busca clasificar si el documento, en su conjunto, expresa un sentimiento positivo o negativo [[3](#_ENREF_3), [5](#_ENREF_5)].

El análisis de sentimientos a nivel de oración profundiza en el análisis de textos, examinando cada frase individualmente. Su objetivo es determinar si cada oración expresa un sentimiento positivo, negativo o neutral [[3](#_ENREF_3), [5](#_ENREF_5)].

Aunque los niveles de documento y oración proporcionan información valiosa, no permiten identificar con precisión qué aspectos específicos generan satisfacción o insatisfacción en los usuarios. En lugar de analizar estructuras lingüísticas como documentos, párrafos u oraciones, este nivel se centra en la propia opinión, identificando el sentimiento y el objeto al que se dirige [[3](#_ENREF_3), [5](#_ENREF_5)].

El creciente interés en el Análisis de Sentimiento Basado en Aspectos (por sus siglas en ingles *ABSA*) en los últimos años ha impulsado la investigación en este campo, especialmente en el idioma inglés. Sin embargo, otros idiomas como el español, chino o checo, entre otros, aún no han recibido la misma atención. El *ABSA* busca extraer los términos de aspectos de una entidad y determinar la polaridad del sentimiento asociado a dicha entidad. Un aspecto se refiere a un atributo o componente específico de la entidad. Por ejemplo, en la frase "La carne de este restaurante es exquisita", el aspecto sería "carne", la entidad sería "restaurante" y el sentimiento asociado sería "exquisito", con una polaridad positiva [[6](#_ENREF_6)].

La extracción de términos de aspecto (por sus siglas en ingles *ATE*) es una subtarea fundamental en el análisis de sentimientos basado en aspectos. Su objetivo principal radica en identificar y extraer los aspectos relevantes presentes en un texto, sin contar con una lista predefinida de dichos aspectos, teniendo en cuenta que los aspectos pueden ser explícitos o implícitos. Esta tarea constituye la etapa inicial del proceso de *ABSA* y se considera una de las más desafiantes debido a su complejidad [[7](#_ENREF_7), [8](#_ENREF_8)].

La *ATE* busca descubrir y capturar los elementos clave que representan los aspectos mencionados en el texto. Estos aspectos pueden variar ampliamente dependiendo del dominio o contexto en el que se aplique. Al realizar una extracción precisa de los términos de aspecto, se facilita enormemente la comprensión y la clasificación del sentimiento expresado en el texto [[7](#_ENREF_7), [8](#_ENREF_8)].

El campo del Procesamiento del Lenguaje Natural (PLN) ha experimentado un avance sin precedentes en las últimas décadas, impulsado en gran medida por la aplicación de métodos de aprendizaje automático. Esta transformación ha permitido desarrollar sistemas capaces de comprender, interpretar y generar lenguaje humano con una precisión y sofisticación nunca antes vistas. Tradicionalmente, el PLN se basaba en reglas gramaticales y análisis lingüísticos, lo que limitaba su capacidad para manejar la complejidad y la ambigüedad del lenguaje natural. Sin embargo, la llegada del aprendizaje automático ha revolucionado la forma en que se aborda el PLN [[9](#_ENREF_9)].

El aprendizaje automático, también conocido como *Machine Learning* (*ML*), ha permitido superar estas limitaciones al permitir que los sistemas aprendan de manera autónoma a partir de grandes conjuntos de datos. Los algoritmos de *ML*, como la regresión lineal, los árboles de decisión y las máquinas de vectores de soporte, han demostrado ser efectivos para tareas como la clasificación de textos, la detección de spam y la extracción de información [[9](#_ENREF_9)].

Sin embargo, el desarrollo del aprendizaje profundo (*Deep Learning*, *DL*) ha impulsado aún más el avance del PLN. Los algoritmos de *DL*, basados en redes neuronales, han demostrado una capacidad excepcional para aprender patrones complejos a partir de grandes cantidades de datos, permitiendo la creación de modelos de lenguaje más precisos y versátiles. Esta capacidad ha sido fundamental para el desarrollo de aplicaciones de *PLN* de vanguardia, como los sistemas de traducción automática, los asistentes virtuales, los *chatbots* de servicio al cliente y los sistemas de análisis de sentimientos [[10](#_ENREF_10), [11](#_ENREF_11)] .

La popularidad del aprendizaje profundo, en particular, ha dirigido la atención hacia arquitecturas de redes neuronales basadas en capas convolucionales y recurrentes para la comprensión y procesamiento del lenguaje. Estas arquitecturas permitieron representaciones más ricas y detalladas en comparación con los modelos tradicionales de *Machine Learning*. Las redes convolucionales, por ejemplo, facilitan el análisis del texto por partes, mediante filtros, mientras que las redes recurrentes procesan el texto secuencialmente, considerando las salidas de partes anteriores como entrada adicional [[11](#_ENREF_11)].

Sin embargo, la aparición de los modelos de redes neuronales basados en la atención, ha reorientado la investigación en aprendizaje profundo para PLN hacia el desarrollo de estructuras transformadoras [[11](#_ENREF_11)].

Estas redes neuronales llamadas *Transformers* son un tipo de modelo de aprendizaje profundo que se utilizan principalmente en el procesamiento del lenguaje natural. Su arquitectura fue propuesta en el trabajo "*Attention Is All You Need*" (2017) [[12](#_ENREF_12)]. La principal característica de las redes neuronales *Transformers* es que utilizan un mecanismo llamado atención para procesar secuencias de datos de longitud variable, como, por ejemplo, frases o párrafos. En lugar de procesar las palabras de forma secuencial, como se hace en los modelos basados en redes neuronales recurrentes (RNN), las redes neuronales *Transformers* pueden procesar todas las palabras en paralelo. Esta capacidad de procesamiento paralelo, junto con el mecanismo de atención, ha llevado a los *Transformers* a alcanzar niveles de precisión y eficiencia nunca antes vistos en el PLN [[11](#_ENREF_11), [12](#_ENREF_12)].

Estos modelos, pre-entrenados con millones de parámetros, requieren solo un ajuste fino para adaptarse a una amplia gama de tareas de *PLN*. Además, requieren menos datos y tiempo de entrenamiento que otros modelos, lo que facilita su implementación y escalabilidad. Su impacto en el *PLN* es notable: son la base de muchos modelos de lenguaje de última generación, como *GPT-4, BERT* y *XLNet*, que impulsan herramientas como *GitHub Copilot*, y la última generación de motores de búsqueda de Google [[10](#_ENREF_10), [11](#_ENREF_11)].

Para mejorar el rendimiento de estos modelos pre-entrenados se estará utilizando una técnica llamada: aprendizaje conjunto, en ingles *Ensemble Learning*, esta es una poderosa estrategia en el aprendizaje automático. Se basa en integrar múltiples modelos de referencia dentro de un mismo marco con el objetivo de crear un modelo "maestro" que supere en rendimiento a cada uno de los modelos individuales. Al combinar las fortalezas de varios modelos, se obtiene un sistema más robusto y preciso. Este enfoque se ha convertido en una tendencia fundamental en la investigación del aprendizaje automático, impulsado por su capacidad de lograr resultados de vanguardia en diversas aplicaciones. El éxito de un método de aprendizaje conjunto depende de dos aspectos clave: el entrenamiento de los modelos base y la combinación de las predicciones [[13](#_ENREF_13), [14](#_ENREF_14)].

Tras una exhaustiva revisión de la literatura, se han detectado dos patrones relevantes en el ámbito del análisis de sentimientos basado en aspectos: primero, la subtarea de extracción de aspectos presenta un desarrollo significativamente menor en comparación con la clasificación de polaridad, con menos investigaciones dedicadas a ella, lo que se traduce en niveles de precisión más bajos en general. Este desequilibrio se acentúa en el contexto del idioma español, donde la escasez de estudios y la consecuente reducción en los niveles de precisión son aún más pronunciadas. Segundo, las técnicas de aprendizaje conjunto han demostrado su eficacia en la mejora del rendimiento de los modelos de análisis de sentimientos. Sin embargo, la aplicación de estas técnicas a la extracción de aspectos en español presenta una carencia notable. No se han encontrado estudios que exploren esta combinación para textos en español, a diferencia de la clasificación de polaridad, donde existen investigaciones que han implementado el aprendizaje conjunto con resultados positivos.

A partir de la situación anterior se plantea como **problema** a resolver la existencia de una baja precisión en la extracción de aspectos para el análisis de sentimientos basado en aspectos en textos en español, especialmente en comparación con la clasificación de polaridad.

El **objeto de estudio** de la presente investigación se enmarca en el análisis de sentimientos. Se identifica como **campo de acción** el análisis de sentimientos basado en aspectos con el uso de modelos *Transformers* y técnicas de aprendizaje conjunto.

El **objetivo general** de esta investigación consiste en desarrollar un algoritmo de análisis de sentimientos basado en aspectos para el idioma español con el uso de técnicas de aprendizaje conjunto en modelos *Transformers* que mejore la precisión en la extracción de aspectos en comparación con los trabajos reportados en la literatura.

Este se desglosa en los siguientes **objetivos específicos:**

1. Caracterizar los fundamentos teóricos y los trabajos relacionados al análisis de sentimientos basado en aspectos.

* Tareas:

1. Estudiar los fundamentos teóricos del análisis de sentimientos que utilicen un enfoque basado en aspectos.
2. Estudiar los fundamentos teóricos de la técnica de aprendizaje conjunto.
3. Identificar trabajos previos relacionados con la extracción de aspectos y clasificación de la polaridad.
4. Seleccionar y estudiar los modelos de *Transformers* para la extracción de aspectos.
5. Desarrollar una nueva solución de análisis de sentimientos basado en aspectos con el uso de modelos *Transformers* y técnicas de ensamble que permita mejorar los resultados de soluciones propuestas en la literatura para idioma español.

* Tareas:

1. Aplicar la limpieza de los textos a utilizar para el aprendizaje del modelo.
2. Concebir un mecanismo de extracción de aspectos.
3. Implementar el método seleccionado.
4. Diseñar una solución para la clasificación de la polaridad.
5. Entrenar el modelo con los datos de entrenamiento
6. Evaluar experimentalmente la solución utilizando colecciones de prueba reconocidos.

* Tarea:

1. Seleccionar las métricas a utilizar para medir la efectividad del modelo.
2. Selección los datos para realizar los experimentos.
3. Efectuar experimentos con el nuevo modelo utilizando las métricas seleccionadas y el conjunto de datos.
4. Comparar los resultados obtenidos del modelo con otros modelos existentes.

El informe está estructurado en tres capítulos:

**Capítulo 1** – Análisis de Sentimientos Basado en Aspectos: Se describen de manera general los conceptos del Procesamiento del Lenguaje Natural y el Análisis de Sentimientos Basado en Aspectos. Se muestran los enfoques más utilizados para la extracción de aspectos y clasificación de la polaridad. Se describen los modelos *Transformers* y se muestran los trabajos relacionados.

**Capítulo 2** - Solución de análisis de sentimientos basado en aspectos para idioma español: En este capítulo se realiza una descripción detalla de cada uno de los procedimientos realizados para dar solución a la problemática planteada. Se dan a conocer cuáles fueron los modelos *Transformers* utilizados. Además, se muestra la documentación asociada a la solución mediante la utilización de diferentes artefactos de la ingeniería de software.

**Capítulo 3** - Evaluación de la solución propuesta: Se muestra cuál fue el marco de evaluación utilizado. Se describen cada una de las posibles configuraciones a realizar para la evaluación. Se comparan los resultados obtenidos en cada uno de los algoritmos propuestos utilizando las métricas de evaluación y se muestra cuál fue el resultado del método propuesto.

# Capítulo 1: Análisis de sentimientos basado en aspectos.

En este capítulo se presentan los conceptos teóricos fundamentales del Procesamiento del Lenguaje Natural (PLN) y el Análisis de Sentimientos Basado en Aspectos (ABSA), con un enfoque especial en la problemática de la extracción de aspectos. Se caracterizan las soluciones más relevantes reportadas en la literatura, enfocadas en el aprendizaje profundo, y se introducen las principales técnicas y métodos empleados, así como sus limitaciones. El capítulo está organizado de la siguiente manera: primero, se abordan los conceptos básicos del PLN; luego, se profundiza en el análisis de sentimientos y su evolución hacia el enfoque basado en aspectos; finalmente, se discuten las técnicas de aprendizaje profundo y los modelos de *Transformers* aplicados a esta tarea.

## Procesamiento del Lenguaje Natural (PLN).

El Procesamiento del Lenguaje Natural (PLN, o PLN por sus siglas en inglés) es un área que combina técnicas computacionales para analizar, entender y generar lenguaje humano de manera automatizada. Su objetivo es lograr que las máquinas procesen información textual de forma similar a como lo haría un ser humano, lo que implica no solo reconocer palabras o sonidos, sino también comprender su significado y contexto [[15](#_ENREF_15), [16](#_ENREF_16)].

El PLN se basa en disciplinas como la informática, la lingüística, las matemáticas, la inteligencia artificial y la psicología, entre otras. Estas contribuyen a desarrollar métodos para representar y manipular textos en diferentes niveles de análisis lingüístico, como el léxico, el sintáctico y el semántico [[15](#_ENREF_15), [16](#_ENREF_16)].

Entre las aplicaciones más comunes del PLN se encuentran [[15](#_ENREF_15)]:

- **Extracción de información**: Identificación y recuperación de datos relevantes en textos.

- **Sistemas de diálogo**: Desarrollo de chatbots y asistentes virtuales capaces de interactuar con usuarios.

- **Análisis de sentimientos**: Determinación de la polaridad (positiva, negativa o neutral) en opiniones o reseñas.

### Niveles de análisis lingüístico en PLN

El Procesamiento del Lenguaje Natural opera en varios niveles de análisis lingüístico, cada uno de los cuales contribuye a la comprensión del lenguaje humano. Estos niveles, que van desde el análisis de sonidos hasta la interpretación del contexto, son fundamentales para desarrollar sistemas de PLN efectivos [[15](#_ENREF_15),[16](#_ENREF_16)].

A continuación, se describen los principales niveles de análisis lingüístico:

* **Fonología:** La fonología se ocupa de la interpretación de los sonidos del habla. En un sistema de PLN que acepta entrada hablada, las ondas sonoras se analizan y codifican en una señal digitalizada para su interpretación.
* **Morfología:** La morfología estudia la naturaleza componencial de las palabras, que están formadas por morfemas, las unidades más pequeñas de significado.
* **Nivel léxico:** En este nivel, tanto los humanos como los sistemas de PLN interpretan el significado de las palabras individuales. Esto incluye:

- **Etiquetado gramatical**: Asignar una etiqueta de parte del discurso (sustantivo, verbo, adjetivo, etc.) a cada palabra basándose en su contexto. - **Representación semántica**: Para palabras con un único significado, se puede reemplazar la palabra por una representación semántica. Por ejemplo, la palabra "lanzar" puede descomponerse en predicados lógicos que representan su significado básico.

* **Nivel sintáctico** El análisis sintáctico se centra en descubrir la estructura gramatical de una oración. Esto implica:

- **Gramáticas y parsers**: Herramientas que identifican las relaciones de dependencia entre las palabras.

- **Estructura de la oración**: Por ejemplo, las oraciones "El perro persiguió al gato" y "El gato persiguió al perro" tienen la misma estructura léxica pero difieren en su sintaxis, lo que cambia su significado.

* **Nivel semántico** Este nivel determina el significado de una oración al analizar las interacciones entre los significados de las palabras. Incluye:

- **Desambiguación semántica**: Seleccionar el sentido correcto de palabras polisémicas (por ejemplo, "archivo" puede referirse a una carpeta o a una herramienta para limar).

- **Representación semántica**: Crear una representación del significado de la oración que integre los significados individuales de las palabras.

* **Nivel de discurso** El análisis del discurso trabaja con unidades de texto más largas que una oración.
* **Nivel pragmático** El nivel pragmático se enfoca en el uso intencional del lenguaje en situaciones específicas.

## Análisis de Sentimientos.

El análisis de sentimiento (por sus siglas en ingles, *SA*) es una de las aplicaciones más populares del PLN. Tiene como objetivo el estudio computacional de opiniones, sentimientos y emociones expresados en texto. Como campo de investigación, está estrechamente relacionada con la lingüística computacional, procesamiento del lenguaje natural y la minería de textos. Se encarga de explorar y descubrir la información subjetiva generada por el usuario. Como campo de estudio, aglomera un conjunto de técnicas para recopilar, agrupar, estructurar, analizar y valorar texto a fin de conocer cómo el ser humano utiliza el lenguaje para expresar información afectiva [[17](#_ENREF_17), [18](#_ENREF_18)].

La investigación en el análisis de sentimientos cubre casi todas las áreas de investigación del procesamiento del lenguaje natural, cubre semántica léxica, análisis semántico, extracción de información, análisis del discurso, entre otras y tiene como objetivo fundamental encontrar las opiniones, identificar sus sentimientos y clasificarlos en opiniones positivas y negativas [[19](#_ENREF_19)].

### Niveles del análisis de sentimientos.

El análisis de sentimientos tiene como componente principal la clasificación de opiniones según su polaridad (positiva, negativa o neutral). Esta clasificación puede realizarse en diferentes niveles, cada uno de los cuales considera una unidad de información distinta: el nivel de documento, el nivel de oración y el nivel de aspecto [5]. A continuación, se describen estos niveles en detalle.

* **Nivel de documento.**

En este nivel, se analiza un documento completo y se le asigna una única polaridad. Este enfoque es útil para clasificar textos extensos, como páginas de libros, blogs o reseñas de productos, donde el contenido refleja una opinión general. Se pueden emplear técnicas de aprendizaje automático, tanto supervisadas como no supervisadas, para realizar esta clasificación. Sin embargo, este método funciona mejor cuando el documento es escrito por una sola persona y no es adecuado para textos que evalúan o comparan múltiples entidades, ya que puede perder matices importantes [[17](#_ENREF_17), [18](#_ENREF_18)].

* **Nivel de Oración.**

A nivel de oración, cada frase se analiza de manera independiente y se le asigna una polaridad específica. Este enfoque utiliza metodologías similares a las del nivel de documento, pero requiere mayores recursos de procesamiento y datos de entrenamiento. Es especialmente útil cuando un documento contiene una mezcla de sentimientos, ya que permite identificar opiniones positivas, negativas o neutrales dentro de un mismo texto. Además, este nivel está asociado con la clasificación subjetiva, lo que lo hace fundamental para tareas como la identificación de oraciones subjetivas, condicionales o ambiguas [[3](#_ENREF_3), [5](#_ENREF_5)].

* **Nivel de Aspecto.**

El análisis de sentimientos a nivel de aspecto es el más granular. En este enfoque, se identifican y analizan los aspectos específicos mencionados en una oración, asignando una polaridad a cada uno de ellos. Posteriormente, se calcula un sentimiento agregado para la oración completa. Este nivel es particularmente útil cuando una oración contiene múltiples aspectos o atributos de una entidad, ya que permite capturar la orientación de la opinión hacia cada uno de ellos de manera individual. Por ejemplo, en la oración "La comida estaba deliciosa, pero el servicio fue terrible", se asignaría una polaridad positiva a "comida" y una negativa a "servicio" [[3](#_ENREF_3), [5](#_ENREF_5)].

## Análisis de sentimientos basado en aspectos.

El Análisis de Sentimientos Basado en Aspectos (ABSA, por sus siglas en inglés) es una técnica avanzada que permite una comprensión más profunda y granular de las opiniones expresadas en un texto [[21](#_ENREF_21)]. A diferencia del análisis de sentimientos tradicional, que clasifica un texto completo como positivo, negativo o neutral, el ABSA se enfoca en identificar y analizar la polaridad del sentimiento hacia características específicas, o aspectos, de un producto, servicio, persona o evento. Un aspecto se define como un atributo relevante que contribuye a la evaluación de un objeto. Por ejemplo, en una reseña de un restaurante, los aspectos podrían ser la comida, el servicio, el ambiente o el precio [[21](#_ENREF_21)].

El ABSA consta de dos tareas principales, cada una de las cuales aborda un componente clave del análisis [[22](#_ENREF_20)]:

1. **Extracción de Aspectos:** Esta etapa consiste en identificar las características o atributos específicos, llamados aspectos, que se mencionan en un texto. La extracción de aspectos es un proceso desafiante, ya que requiere comprender el contexto del texto y reconocer los elementos relevantes que se están evaluando. Esta tarea es fundamental para garantizar que el análisis de sentimientos sea preciso y significativo.

2. **Clasificación de Sentimientos:** Una vez identificados los aspectos, el siguiente paso es determinar la polaridad del sentimiento expresado hacia cada uno de ellos. Esto implica clasificar la opinión sobre el aspecto como positiva, negativa o neutral. Para lograrlo, se analiza el contexto lingüístico alrededor del aspecto, lo que permite inferir la intención del autor. Por ejemplo, en la oración "La comida del restaurante es excelente, pero el servicio es malo", se asignaría una polaridad positiva al aspecto "comida" y una polaridad negativa al aspecto "servicio".

## Enfoques utilizados en el ABSA

El análisis de sentimientos basado en aspectos ha experimentado un auge significativo en los últimos años, impulsado por la necesidad de comprender las opiniones de los usuarios con mayor profundidad y granularidad. Este campo de investigación ha dado lugar a una variedad de enfoques, cada uno con sus propias fortalezas y limitaciones. Entre los métodos más utilizados se encuentran los modelos basados en redes neuronales, las técnicas de aprendizaje profundo y los modelos de Transformers. Estos enfoques pueden clasificarse en dos categorías principales: supervisados y no supervisados, cuya elección depende de factores como la disponibilidad de datos etiquetados, la complejidad de la tarea y los objetivos específicos de la investigación [[23](#_ENREF_23)].

### Enfoque no supervisado.

El análisis de sentimientos basado en aspectos (ABSA) no supervisado se caracteriza por su capacidad para extraer aspectos y clasificar la polaridad del sentimiento sin depender de conjuntos de datos pre-etiquetados. Estas técnicas se basan en la identificación de patrones y relaciones dentro del texto para inferir la estructura de la opinión, lo que las hace especialmente útiles en escenarios donde la anotación manual es inviable o costosa [[21](#_ENREF_21)].

La investigación pionera de [[24](#_ENREF_21)] sentó las bases para el enfoque no supervisado en el resumen de sentimientos basado en aspectos. Este trabajo propuso un método para la detección de aspectos mediante el análisis de patrones basados en sustantivos, lo que permite extraer información relevante sobre los diferentes atributos de un producto o servicio. Este enfoque fue fundamental para el desarrollo de técnicas no supervisadas en el ABSA, abriendo camino a nuevas investigaciones en la extracción de aspectos y el resumen de opiniones.

En [[25](#_ENREF_21)] se propone un enfoque basado en ontologías para la extracción de aspectos, utilizando el recurso lingüístico ***SentiWordNet*** para la clasificación de polaridad. Este método destaca por su capacidad para considerar palabras negativas en el análisis, lo que mejora la precisión en la detección de sentimientos. La incorporación de ontologías y *SentiWordNet* permite una comprensión más profunda del contexto y las relaciones semánticas entre las palabras, lo que resulta en una mejor interpretación de la polaridad de los comentarios. Este enfoque representa un avance significativo al abordar la complejidad de las expresiones negativas, que a menudo son difíciles de procesar con métodos tradicionales.

Por otro lado, [[26](#_ENREF_21)] presenta un enfoque prometedor para la extracción de aspectos mediante la combinación de ***Latent Dirichlet Allocation (LDA)*** con un modelo pre-entrenado no supervisado. Este método se destaca por su independencia lingüística y contextual, un atributo poco común en el campo del análisis de sentimientos. Sin embargo, el sistema propuesto tiene limitaciones, ya que requiere intervención manual para proporcionar información sobre palabras relacionadas con el dominio específico bajo análisis. Esta dependencia introduce un riesgo de error humano y afecto la consistencia de los resultados. Futuras investigaciones podrían explorar alternativas que utilicen aprendizaje automático para la detección automática de términos relevantes, reduciendo así la necesidad de intervención humana y aumentando la precisión y robustez del análisis.

### Enfoque supervisado.

El aprendizaje supervisado en el ABSA se basa en la construcción de una relación entre variables de entrada (X) y una variable de salida (Y). Esta relación se aprende a partir de datos etiquetados, donde cada ejemplo está asociado a una etiqueta que indica la clase o valor deseado. El objetivo es utilizar esta relación aprendida para predecir la salida de nuevos datos no vistos [[27](#_ENREF_27)].

La esencia del aprendizaje supervisado reside en su capacidad para generar predicciones precisas sobre datos futuros basándose en el análisis de información previamente etiquetada. Este proceso implica dos fases principales [[27,28](#_ENREF_27)]:

1. **Fase de entrenamiento**: Los algoritmos aprenden la relación entre las variables de entrada y salida utilizando datos etiquetados.

2. **Fase de prueba**: Se evalúa la precisión de las predicciones sobre datos nuevos no vistos durante el entrenamiento.

Aunque muchos trabajos de investigación han utilizado algoritmos supervisados tradicionales de *Machine Learning* para el ABSA, este epígrafe se enfoca en las **redes neuronales profundas**, que han demostrado ser herramientas más sofisticadas y efectivas para esta tarea. Entre las arquitecturas más destacadas se encuentran [[29](#_ENREF_27)]:

- **Redes Neuronales Recurrentes (RNN)**: Ideales para capturar dependencias temporales en secuencias de texto.

- **Redes Neuronales Convolucionales (CNN)**: Efectivas para extraer características locales en textos.

- **Transformers**: Modelos basados en atención que capturan el contexto global del texto.

En [[20](#_ENREF_20)] se propone un nuevo modelo para la extracción de aspectos basado en el aprendizaje profundo y permanente, utilizando una técnica de reducción catastrófica del olvido en múltiples dominios. La arquitectura se basa en una CNN y un modelo de *Lifelong Learning*, compuesto por cuatro etapas. En la primera etapa, las opiniones textuales se convierten en vectores *Word Embedding*. La segunda etapa, la extracción de conocimientos básicos, entrena la CNN para cada dominio y actualiza la Base de Conocimiento (KB). La etapa tres, la actualización de la KB, utiliza un análisis de pérdidas para evitar el olvido catastrófico, enriqueciendo la KB con nuevos aspectos extraídos de los diferentes dominios. Finalmente, la etapa cuatro crea el extractor de aspectos, un modelo CNN multidominio que se configura con los parámetros comunes de la KB.

Del mismo modo, [[21](#_ENREF_21)] propone un modelo llamado BERT *Adversarial* Training (BAT) para mejorar la extracción de aspectos y la clasificación de sentimientos de aspectos en análisis de sentimientos. El modelo utiliza un entrenamiento adversario para ajustar el BERT de propósito general y el BERT post-entrenado específico de dominio (BERT-PT). El entrenamiento adversario implica crear ejemplos contradictorios a partir de las incrustaciones de BERT utilizando el gradiente de pérdida, y luego alimentar estos ejemplos perturbados al codificador BERT para calcular la pérdida adversaria. Finalmente, se aplica el algoritmo de retropropagación a la suma de la pérdida de entrenamiento y la pérdida adversaria. Los experimentos realizados muestran que BAT supera al BERT general y al BERT post-entrenado en ambas tareas, tanto en el ámbito de restaurantes como de portátiles.

También en [[30](#_ENREF_30)] se propone un marco de aprendizaje conjunto basado en BERT para abordar todas las subtareas del ABSA. En lugar de tratar cada subtarea por separado, este marco utiliza dos modelos BERT-MRC (*Machine Reading Comprehension*) con parámetros compartidos para resolver las tareas de extracción de aspectos y clasificación de polaridad de sentimiento. Los modelos se entrenan conjuntamente para aprender a identificar las posiciones inicial y final de los aspectos y opiniones en una oración, así como la polaridad del sentimiento expresado.

## Trabajos relacionados.

Este epígrafe presenta en la una revisión exhaustiva de los trabajos relacionados en el ámbito del análisis de sentimientos basado en aspectos utilizando modelos de lenguaje de aprendizaje profundo.

Tabla 1. Soluciones de ABSA reportadas en la literatura

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Referencia** | **Idioma** | **Método** | **Conjunto de datos** | **Extracción de aspectos** | **Polaridad** |
| [[20](#_ENREF_20)] | Ingles | Convolucional CNN | Amazon y tripAdvisor | Restaurante: 62,5  Hotel. 67,9 | Restaurante: 74,8  Hotel: 78,6 |
| [[31](#_ENREF_31)] | Ingles | T5 | ALSC (Rest16 y MAMS). | 79.71 | -- |
| [[32](#_ENREF_32)] | Ingles | Wo-BiLSTM-CRF-Glove.42B | SemeEval-14 | 86,05 | 81,08 |
| [[33](#_ENREF_33)] | Italiano | RoBERTa | Plataforma de comercio electrónico. | 61.87 | -- |
| [[34](#_ENREF_34)] | hindi | Attention-based BiLSTM | Opiniones de artículos electrónicos | 43.16 | -- |
| [[21](#_ENREF_21)] | Ingles | BERT Adversarial Training (BAT) | SemEval 2014 (Task 4) | Laptop: 85.57  Restaurante:81,5 | -- |
| [[35](#_ENREF_35)] | Ingles | BERT post-training | SemEval 2014 (task 4) | Laptop: 84.26  Restaurante: 77,9 | -- |
| [[36](#_ENREF_36)] | Ingles | DomBERT | SemEval 2014 (Task 4) | Rest : 77.21 | -- |
| [[37](#_ENREF_37)] | Ingles | BART | SemEval 2014 - 2015 | Rest 14: 87.07  Rest 15: 75.48  Lapt 14: 83.52 | -- |
| [[38](#_ENREF_38)] | Ingles | BERT-MTL-  GRU | SemEval 2014 | Restaurante::86.19 | -- |
| [[30](#_ENREF_30)] | Ingles | Bert- Dual-MRC | SemEval 2014 - 2015 | Rest 14: 86.60  Rest 15: 75.08  Lapt 14: 82.51 | -- |
| [[39](#_ENREF_39)] | Español | OTE: Ontologías del dominio  ASC:PMI | Semeval 2016 | F1: 73,07 | Acc: 84.80 |
| [[40](#_ENREF_40)] | Español | Convolucional (CNN) | Semeval 2016 | F1: 65,40 | Acc:79.69 |
| [[22](#_ENREF_22)] | Español | OTE: Bert  ASC: RoBERTa + WW(size 5) | Semeval 2016 | Prec: 73.60  Rec: 81.50  F1: 76.70 | Acc: 86.60  Prec: 83.60  Rec: 86.40  F1: 84.80 |
| [[28](#_ENREF_22)] | Español | No reportado | Semeval 2016 | F1: 68.50 | Acc: 83.50 |
| [[6](#_ENREF_22)] | Español | OTE: Ontologías del dominio  ASC: SVM+Lexicón | Semeval 2016 | F1:73.00 | Acc:84.70 |
| [[58](#_ENREF_22)] | Español | Bi-LSTM | Semeval 2016 | F1:73.00 | Acc: 87.10 |
| [[59](#_ENREF_22)] | Español | BiLSTM-CRF | Semeval 2016 | F1: 71.70 | ---- |

## Redes neuronales Transformers.

Las redes neuronales *Transformers* han transformado el campo del procesamiento del lenguaje natural. Su arquitectura innovadora, basada en un mecanismo de atención llamado "autoatención" (*self-attention*), les permite procesar secuencias de datos de longitud variable, como frases o párrafos, de manera más eficiente que los modelos tradicionales basados en redes neuronales recurrentes (RNN) [[12](#_ENREF_12)].

En lugar de procesar las palabras secuencialmente, como en los RNN, los *Transformers* pueden procesar todas las palabras en paralelo, lo que los hace mucho más rápidos y efectivos. Esta capacidad de capturar relaciones complejas entre las palabras a través de la atención les permite generar representaciones contextuales más ricas y precisas [[12](#_ENREF_12)].

La arquitectura de un *Transformers* se basa en bloques de atención que se apilan en capas. Cada bloque *Transformers* se compone de dos subcapas principales [[12](#_ENREF_12), [22](#_ENREF_22)]:

1. Capa de Atención: En esta capa, se calculan los pesos de atención para cada palabra en la secuencia, ponderando su importancia en función de su relación con las demás palabras. Este proceso se realiza mediante cálculos de producto escalar y una función *Softmax*, que permite que las palabras más relevantes tengan un mayor peso en la representación final.

2. Capa de Alimentación hacia Adelante (*Feed-forward*): Esta capa aplica una transformación lineal y una función de activación no lineal a la salida de la capa de atención, agregando más complejidad y expresividad a la representación de la secuencia.

El proceso de entrenamiento de un *Transformer* utiliza la técnica de retropropagación del error. Esto permite ajustar los pesos de la red neuronal de manera que la salida generada por el modelo se acerque lo más posible a la salida deseada para la tarea en cuestión [[12](#_ENREF_12), [22](#_ENREF_22)].

Este tipo de redes presenta ventajas significativas con respecto a otros tipos de redes neuronales [[8](#_ENREF_8), [22](#_ENREF_22)].

* Capacidad para procesar secuencias largas de texto: Como se mencionaba anteriormente, a diferencia de las arquitecturas tradicionales de redes neuronales recurrentes, que sufren de limitaciones en el procesamiento de secuencias largas, las redes *Transformer* pueden manejar textos de longitud variable con una mayor eficiencia. Esto se debe a su mecanismo de atención, que permite a la red enfocarse en partes específicas de la secuencia de entrada, en lugar de procesarla secuencialmente. Esta capacidad es crucial para el análisis de sentimientos basado en aspecto, ya que las expresiones de sentimientos y opiniones a menudo se distribuyen a lo largo de textos extensos, y la comprensión de la relación entre palabras y frases distantes es fundamental para interpretar el sentimiento general.
* Captura de relaciones a largo plazo entre palabras: Las redes *Transformer* son capaces de capturar relaciones a largo plazo entre las palabras en una secuencia, lo que les permite comprender mejor el contexto y producir resultados más precisos. Este es un aspecto fundamental para el análisis de sentimientos, ya que las opiniones y emociones a menudo se expresan de manera sutil y a través de conexiones indirectas entre palabras en el texto. Por ejemplo, en una frase como "El hotel era bonito, pero la comida no era buena", la red *Transformer* puede comprender que "bonito" se relaciona con "hotel" y "no era buena" se relaciona con "comida", permitiendo una interpretación precisa del sentimiento expresado.
* Escalabilidad y aprendizaje adaptativo: Las redes *Transformer* son altamente escalables y pueden entrenarse con grandes conjuntos de datos, lo que les permite obtener resultados altamente precisos en tareas complejas de PLN. Al ser expuestas a una amplia gama de datos, las redes *Transformer* pueden aprender a identificar patrones y relaciones sutiles en el lenguaje, mejorando su capacidad para interpretar sentimientos expresados en diferentes contextos.
* Transferencia de aprendizaje: Son especialmente útiles para la transferencia de aprendizaje, es decir, la capacidad de utilizar el conocimiento aprendido en una tarea para mejorar el rendimiento en otra tarea relacionada. Esto se debe a que las redes neuronales *Transformers* aprenden representaciones de palabras altamente generalizadas que pueden aplicarse a una amplia variedad de tareas de PLN.

## Modelos pre-entrenados de lenguaje con arquitectura Transformers.

La arquitectura *Transformers* ha revolucionado el campo del Procesamiento del Lenguaje Natural al integrar las mejores prácticas existentes y añadir innovaciones significativas. La inclusión de la atención multicabeza permite al modelo comprender mejor la estructura gramatical del lenguaje, más allá de la simple semántica de las palabras, lo que lo convierte en un modelo más potente para la resolución de problemas de PLN. La influencia del *Transformer* es tan profunda que la mayoría de los modelos de vanguardia en PLN, incluyendo modelos de lenguaje basados en *Transformers*, se basan en sus principios para lograr resultados sobresalientes en tareas de clasificación, generación y comprensión de texto [[29](#_ENREF_29)].

### BERT (*Bidirectional Encoder Representation from Transformers*).

BERT es un modelo de lenguaje pre-entrenado que se destaca por su capacidad de comprender el significado de las palabras dentro del contexto de una oración o párrafo completo. Esta arquitectura, basada en las redes neuronales *Transformers*, utiliza un mecanismo de atención para capturar la relevancia de las palabras en el texto, permitiendo una interpretación más profunda del lenguaje [[41](#_ENREF_41)].

Para entrenar a BERT, se empleó una gran cantidad de datos textuales y se implementaron dos técnicas de aprendizaje auto-supervisado. La primera técnica consiste en predecir palabras faltantes en una oración a partir del contexto circundante, conocida como "máscara de palabras". La segunda técnica consiste en predecir la oración siguiente en un texto. Estas técnicas de auto-supervisión permiten que BERT aprenda a representar el lenguaje de forma más eficiente y precisa [[41](#_ENREF_41), [42](#_ENREF_42)].

BERT no se limita a ser un simple mecanismo de representación textual; es un modelo de lenguaje robusto que, además de generar representaciones de palabras, puede utilizarse como base para una amplia gama de tareas de procesamiento del lenguaje natural, como la clasificación de texto, la extracción de información, el análisis de sentimientos y muchas otras [[41](#_ENREF_41), [43](#_ENREF_43)].

Para aplicar BERT a tareas para las que no fue específicamente entrenado, se utiliza el aprendizaje por transferencia. Este proceso implica tomar el modelo pre-entrenado de BERT, que ha aprendido a predecir la oración siguiente y a completar palabras faltantes, y ajustar finamente sus pesos para adaptarlo a una tarea específica [[43](#_ENREF_43)].

Este proceso de ajuste fino (*fine-tuning*) permite que BERT alcance un alto nivel de precisión en diversas tareas de procesamiento del lenguaje natural. La incorporación de una capa adicional de neuronas a la salida de BERT puede mejorar aún más su rendimiento, lo que permite obtener resultados de vanguardia sin necesidad de modificaciones significativas en la arquitectura del modelo original [[41](#_ENREF_41)].

### ALBERT (*A Lite BERT*).

El modelo ALBERT es una variante del modelo BERT, se presenta como una alternativa más eficiente para el procesamiento del lenguaje natural. Si bien BERT se destaca por su capacidad para comprender el significado de las palabras en contexto, su tamaño y complejidad computacional limitan su aplicación en algunos escenarios. ALBERT aborda este problema al reducir la cantidad de parámetros y mejorar la eficiencia del entrenamiento. ALBERT se basa en los principios de BERT, pero introduce mejoras que optimizan su arquitectura. Su estructura se compone de las siguientes capas [[44](#_ENREF_44)]:

* Capa ALBERT: Esta capa es la base del modelo y utiliza la misma técnica de atención multicabeza que BERT, pero con un número reducido de parámetros.
* Capa LSTM (*Long Short-Term Memory*): Esta capa se encarga de capturar la información secuencial del texto, lo que permite comprender el contexto de las palabras y frases.
* Capa de Discriminación de Tendencia Emocional: Esta capa final realiza la clasificación de la polaridad emocional del texto, es decir, determina si el sentimiento expresado es positivo, negativo o neutral.

### RoBERTa (*Robustly optimized BERT approach*).

RoBERTa es una versión mejorada del modelo BERT que se basa en un entrenamiento más robusto y exhaustivo para lograr un mejor rendimiento en tareas de procesamiento del lenguaje natural [[45](#_ENREF_45)].

En lugar de entrenar con un solo conjunto de datos como BERT, RoBERTa se alimenta con un conjunto de datos mucho más amplio, incluyendo *CC-news* (76GB), *OpenWebText* (38GB) y *Stories* (31GB), lo que le permite aprender de una mayor diversidad de lenguaje. Además, RoBERTa se entrena durante más tiempo y con un tamaño de lote más grande, lo que permite que el modelo procese más información y converja a una solución óptima más rápidamente. También se eliminó la tarea de predicción de la siguiente oración del entrenamiento, lo que permite al modelo enfocarse en otras tareas de comprensión del lenguaje, como la comprensión contextual y la predicción de palabras faltantes [[45](#_ENREF_45), [46](#_ENREF_46)].

RoBERTa utiliza un patrón de enmascaramiento diferente al de BERT, lo que mejora su capacidad de comprensión del lenguaje. Esto permite al modelo generar representaciones más precisas y versátiles, lo que se traduce en un mejor rendimiento en una variedad de tareas de procesamiento del lenguaje natural. RoBERTa está disponible en versiones multilingües, incluyendo español. Algunos ejemplos de modelos pre-entrenados para español incluyen BERTIN, RoBERTa-base y RoBERTa-large. Aunque todos comparten la misma arquitectura, sus diferencias residen en los conjuntos de datos utilizados para su entrenamiento y los mecanismos de entrenamiento específicos [[45](#_ENREF_45), [46](#_ENREF_46)].

### GPT-2 (*Generative Pre-trained Transformer 2*).

El modelo GPT-2, desarrollado por *OpenAI*, es un modelo de lenguaje basado en Transformers capaz de generar secuencias largas de texto, adaptándose al estilo y contenido de una entrada arbitraria. Además, los modelos GPT-2 pueden realizar diversas tareas de PLN, como la clasificación (que es el enfoque de este trabajo) [[47](#_ENREF_47)].

Con 1.5 billones de parámetros, GPT-2 está entrenado para predecir la siguiente palabra, dados todas las palabras previas dentro de un texto. Su arquitectura implementa una red neuronal profunda, donde las entradas son procesadas por múltiples capas de neuronas que activan las capas subsecuentes, generando la salida del modelo en la capa final. Este modelo de lenguaje utiliza la arquitectura Transformer, beneficiándose del mecanismo de atención, lo que permite al modelo enfocarse selectivamente en las piezas de entrada más relevantes [[29](#_ENREF_29), [47](#_ENREF_47)].

La arquitectura *Transformers* de GPT-2 tiene dos modelos dispuestos linealmente, el codificador y el decodificador. El primero está compuesto por una capa de autoatención que aplica el mecanismo de atención al texto de entrada y un tipo de avance (*feed-forward)* que convierte el resultado de la capa de autoatención en una incrustación de menor longitud. El segundo también está compuesto por estas unidades, pero con una capa adicional llamada atención codificador-decodificador, la cual mapea el resultado de la capa de autoatención del decodificador con las incrustaciones producidas por el codificador. De esta manera, todo el texto de entrada se procesa a la vez, proporcionando una memoria más estructurada para manejar dependencias a largo plazo en el texto, contrariamente a las técnicas previas que involucraban mecanismos recurrentes [[29](#_ENREF_29), [47](#_ENREF_47)].

Aunque tanto BERT como GPT-2 se basan en la arquitectura *Transformer*, GPT-2 consiste en bloques de decodificador apilados, mientras que BERT está construido utilizando los bloques de codificador. Al igual que los modelos de lenguaje tradicionales, GPT-2 recibe los vectores de palabras y proporciona las estimaciones para la probabilidad de la siguiente palabra, donde cada palabra tiene el contexto de los tokens previos. Después de la generación de cada token, se agrega a la secuencia de entradas, convirtiéndose en la entrada del modelo en su siguiente paso. BERT, sin embargo, no es autorregresivo, utilizando todo el contexto circundante a la vez. Además, en lugar de una capa de autoatención como BERT, GPT-2 utiliza una autoatención enmascarada, la cual enmascara los tokens futuros bloqueando información de los tokens a la derecha de la posición que se está calculando [[29](#_ENREF_29), [47](#_ENREF_47)].

### ELECTA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately).

ELECTRA representa un avance significativo en el pre-entrenamiento de modelos de lenguaje, introduciendo una nueva estrategia basada en la teoría de juegos. En lugar de utilizar la función de pérdida tradicional del modelo BERT, que se basa en la predicción de tokens enmascarados (*Masked Language Model - MLM*), ELECTRA propone una función de pérdida adversarial [[22](#_ENREF_22)].

Este enfoque adversarial ofrece varias ventajas sobre los métodos tradicionales. En primer lugar, el entrenamiento se realiza sobre todos los tokens de entrada, no solo sobre los enmascarados. Esto permite que el modelo aprenda de manera más eficiente. En segundo lugar, la competencia entre el generador y el discriminador fomenta la generación de representaciones contextuales más robustas [[48](#_ENREF_48)].

ELECTRA introduce un cambio significativo en la función de pérdida, pero no altera la arquitectura del *Transformer.* Sin embargo, la arquitectura del *Transformer* presenta una redundancia secuencial que puede aumentar el coste del pre-entrenamiento. *Funnel Transformer* aborda este problema mediante una nueva arquitectura que reduce la redundancia. El enfoque de *Funnel Transformer* se basa en la idea de comprimir la secuencia completa de estados ocultos en el codificador utilizando una técnica de agrupación. Luego, se recupera la representación completa de la secuencia en el decodificador mediante una técnica de muestreo ascendente [[48](#_ENREF_48)].

### Comparación de las arquitecturas de los modelos utilizados

A continuación, en la , se resume las características clave de los modelos *Transformers* empleados, destacando diferencias en parámetros, capas, dimensiones de embeddings y cabezas de atención.

Tabla 2. Comparación de las arquitecturas de los modelos utilizados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelos | Parámetros  (millones) | Capas | Embeddings | Cabezas de atención |
| Bert | 167 | 12 | 768 | 12 |
| Beto (versión de Bert para español) | 109 | 12 | 768 | 12 |
| Albert\_base | 109 | 12 | 768 | 12 |
| Albert\_large | 335 | 24 | 1024 | 16 |
| Albert\_xx\_large | 2 562 | 12 | 4096 | 64 |
| Bertin\_base (versión de RoBerta para español) | 124 | 12 | 768 | 12 |
| Bertin\_large (versión de RoBerta para español) | 355 | 24 | 1024 | 16 |
| Electra\_small | 17 | 12 | 256 | 4 |
| Electra\_base | 109 | 12 | 768 | 12 |
| GPT\_2 | 124 | 12 | 768 | 12 |

## Técnicas de ensamble de clasificadores.

La técnica de ensamble o aprendizaje conjunto, se refiere a métodos que integran múltiples modelos de referencia en el mismo marco para obtener un modelo más fuerte. El éxito de un método de ensamble depende de varios factores, incluido como se entran los modelos de referencia y como se combinan sus predicciones [[49](#_ENREF_49)].

Las técnicas de aprendizaje conjunto han logrado un rendimiento de vanguardia en diversas aplicaciones de aprendizaje automático al combinar las predicciones de dos o más modelos base. La idea fundamental detrás del aprendizaje conjunto es el reconocimiento de que los modelos base tienen limitaciones y cometen errores. Por lo tanto esta técnica tiene como objetivo mejorar el rendimiento de la clasificación aprovechando las fortalezas de múltiples modelos base. Los ensambles de clasificadores se pueden dividir en dos grupos: los que usan el mismo clasificador entrenado varias veces, llamados ensambles homogéneos y los que utilizan diferentes clasificadores, llamados heterogéneos [[14](#_ENREF_14)]. A continuación se explican los métodos de ensamble más utilizados.

### Método de votación (*Voting Ensemble*).

Los métodos de votación se utilizan generalmente en problemas de clasificación o regresión para mejorar el rendimiento predictivo. Estos métodos se dividen en tres categorías: votación máxima, votación promedio y votación promedio ponderada [[13](#_ENREF_13)].

### Votación máxima.

El primer y más popular método de votación es la votación máxima [[50](#_ENREF_50)], a menudo conocido como votación mayoritaria o votación dura. La idea de este método implica recopilar predicciones para cada etiqueta de clase y predecir la etiqueta de clase con la mayor cantidad de votos. Suponiendo que combinamos tres clasificadores, C1, C2, C3 que asignan las siguientes clasificaciones a una muestra de entrenamiento: [0, 0, 1], el método de votación máxima estaría dando como resultado [0, 0, 1] = 0 ya que fue la de mayor voto. Este método tiene la ventaja de ser fácil de entender y es el método más sencillo de implementar [[13](#_ENREF_13), [51](#_ENREF_51)]. En la figura 1 se puede ver el flujo de votación por mayoría

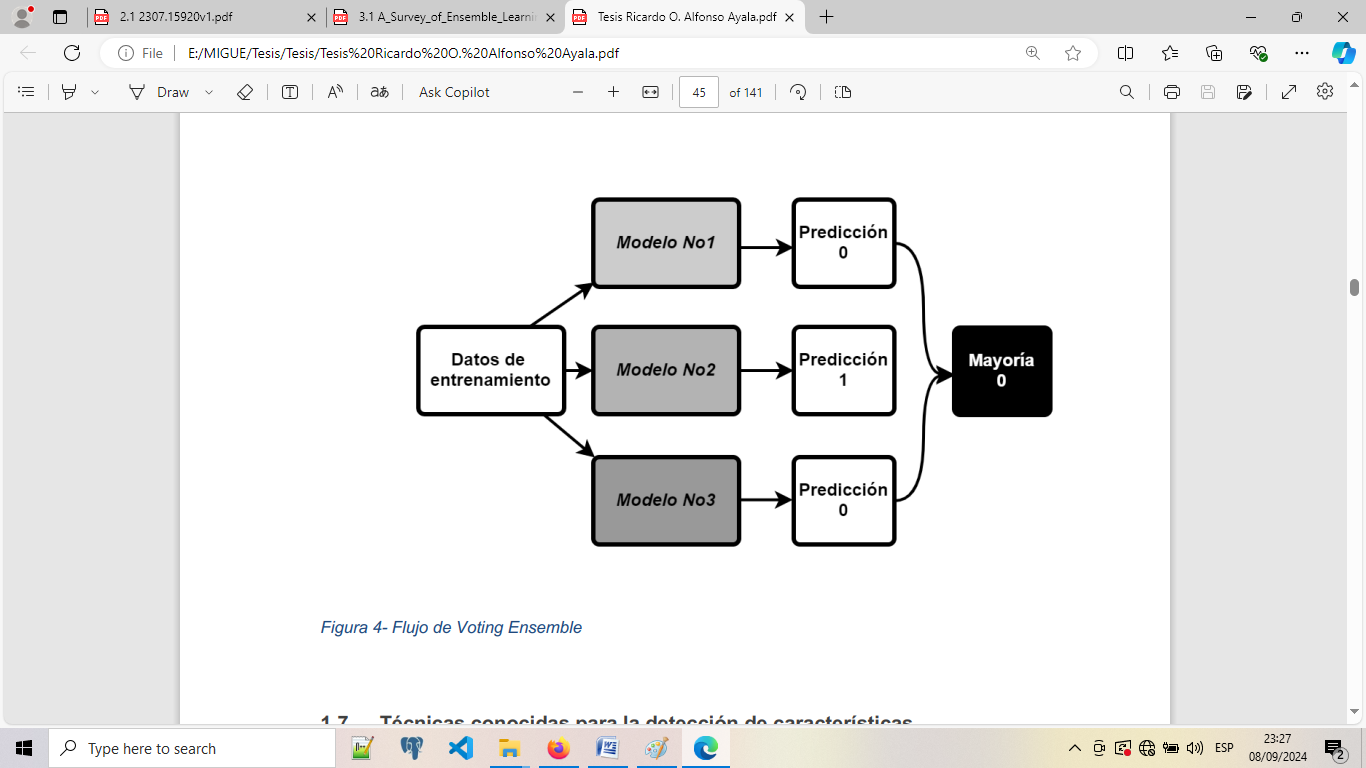


Figura 1. Flujo de ensamble por votación máxima.

### Votación promedio.

El segundo método de votación es la votación promedio [[52](#_ENREF_52)]. La idea de promediar la votación es que las predicciones se extraen de múltiples modelos y se utiliza un promedio de las predicciones para hacer la predicción final. La predicción promedio se calcula utilizando la media aritmética, que es la suma de las predicciones dividida por el total de predicciones realizadas. Por ejemplo, supongamos que el conjunto de clasificadores contiene tres miembros: C1(x)= [0.9, 0.1], C2(x)= [0.2, 0.8] y C3(x)= [0.6, 0.4]. La predicción media sería la siguiente: para calcular la clase 0, (0,9 + 0,2 + 0,6) / 3 = 0,566. Y calcular la clase 1, (0,1 + 0,8 + 0,4) / 3 = 0,433 arrojaría una predicción y = 0. El método de votación promedio tiene la ventaja de ser el más fuerte desde el punto de vista del poder predictivo. Además, su rendimiento es más preciso que la votación por mayoría y reduce el sobreajuste. Las desventajas del método de votación promedio incluyen que es computacionalmente más costoso que el método de votación máxima, ya que requiere promediar los resultados de predicción de todos los modelos de referencia [[13](#_ENREF_13)].

### Votación promedio ponderada.

El tercer método de votación es la votación promedio ponderada, que es una versión ligeramente modificada del voto promedio [[53](#_ENREF_53)]. La idea de la votación promedio ponderada consiste en otorgar diferentes pesos a los alumnos de referencia, lo que indica la importancia de cada modelo en la predicción. Después se multiplica cada predicción por el peso de los clasificadores para producir una suma ponderada y luego dividiendo el resultado por la suma de los pesos del clasificador, estos pesos se pueden usar para calcular el promedio ponderado para cada clase 0 o clase 1. Por ejemplo, supongamos que el conjunto de clasificadores contiene tres miembros: C1(x)= [97.2, 2.8], C2(x)= [100.0, 0] y C3(x)= [95.8, 4.2]. Tiene pesos constantes para los miembros del conjunto [0,84, 0,87, 0,75]. Para calcular la clase 0 = ((97.2 \* 0.84) + (100.0 \* 0.87) + (95.8 \* 0.75))/ (0.84 + 0.87 + 0.75) = 97.763. Y para calcular la clase 1 y 1 = ((2.8 \* 0.84) + (0 \* 0.87) (4.2 \* 0.75))/ (0.84 + 0.87 + 0.75) = 2.235, se obtendría una predicción y = 0. Este método de votación es más preciso que el método de votación promedio simple. El desafío al utilizar un conjunto promedio ponderado es elegir la ponderación relativa de cada miembro. Además, el cálculo es más caro que el método de votación promedio, ya que requiere calcular el promedio ponderado de los resultados de predicción de todos los modelos de referencia, lo que lo hace de poca aplicación [[13](#_ENREF_13)].

### Método de metaaprendizaje.

El metaaprendizaje, también conocido como "aprender a aprender", es un proceso que implica aprender de los propios aprendizajes [[54](#_ENREF_54)]. Este término abarca el aprendizaje basado en experiencias previas con otras tareas. Se utiliza para mejorar el rendimiento y los resultados de un algoritmo de aprendizaje, ajustando ciertos aspectos del algoritmo según los resultados de los experimentos. El metaaprendizaje se diferencia de los modelos tradicionales de aprendizaje automático al incorporar múltiples etapas de aprendizaje. Los resultados de los inductores individuales sirven como entrada para un metaaprendiz, que finalmente genera el resultado final. En los últimos años, el interés en el metaaprendizaje ha aumentado considerablemente, especialmente después de 2017. El auge de algoritmos avanzados de aprendizaje automático ha planteado nuevos desafíos para su entrenamiento, lo que ha impulsado la investigación en metaaprendizaje. Estos algoritmos enfrentan dificultades como altos costos operativos debido a la gran cantidad de experimentos durante la fase de entrenamiento, y la dificultad de encontrar el modelo óptimo para un conjunto de datos determinado [[13](#_ENREF_13), [51](#_ENREF_51)]

### Método de embolsado (*Bagging*).

El método de bagging [[20](#_ENREF_20)], es un algoritmo completamente específico de los datos. Se basa en la creación de múltiples subconjuntos de datos a partir del conjunto de datos original. El objetivo del bagging es generar modelos predictivos más diversos ajustando la distribución estocástica de los conjuntos de entrenamiento, de manera que pequeñas variaciones en el conjunto de entrenamiento produzcan cambios significativos en las predicciones del modelo. Bagging es una abreviatura de la combinación de bootstrapping y agregación. En el bootstrapping, el entrenamiento de los modelos del conjunto se realiza sobre réplicas bootstrap del conjunto de entrenamiento. En la agregación, el resultado final se obtiene mediante una votación mayoritaria de las predicciones de los modelos para determinar la predicción fina [[13](#_ENREF_13)].

El bagging ofrece la ventaja de reducir la varianza, lo que ayuda a evitar el sobreajuste. También funciona bien con datos de alta dimensionalidad. La desventaja del bagging es que es computacionalmente costoso y tiene un alto sesgo, además de que puede dificultar la interpretabilidad del modelo. Un buen ejemplo de bagging es el algoritmo de Bosques Aleatorios [[13](#_ENREF_13)].

### Método de apilamiento (*Stacking*).

El método de *stacking*, es una técnica de ensamblaje de modelos que se utiliza para combinar información de múltiples modelos predictivos y generar un nuevo modelo (meta-modelo). La arquitectura de un modelo de stacking involucra dos o más modelos base, conocidos como modelos de nivel 0, y un meta-modelo que combina las predicciones de los modelos base, conocido como modelo de nivel 1. En los modelos de nivel 0 (modelos base), los modelos se ajustan a los datos de entrenamiento y sus predicciones se compilan. Sin embargo, en el modelo de nivel 1 (meta-modelo), el modelo aprende cómo combinar las predicciones de los modelos base de la mejor manera. Las salidas de los modelos base que se utilizan como entrada para el meta-modelo pueden ser valores de probabilidad o etiquetas de clase en el caso de la clasificación [[13](#_ENREF_13)].

El método de stacking generalmente supera el rendimiento de todos los modelos entrenados individualmente. El sobreajuste es un problema importante con el stacking de modelos, ya que hay muchos predictores que predicen el mismo objetivo y se fusionan. Además, el stacking de múltiples niveles es costoso en términos de datos (se necesita una gran cantidad de datos para el entrenamiento) y de tiempo (ya que cada capa agrega múltiples modelos) [[13](#_ENREF_13)].

### Método de reforzamiento (*Boosting*).

El método de **boosting** fue introducido por primera vez por Freund y Schapire en 1997. Es un proceso secuencial en el que cada modelo subsiguiente intenta corregir los errores del modelo anterior. Boosting consiste en entrenar múltiples **aprendices débiles** (modelos simples) de manera adaptativa, donde cada modelo en la secuencia se ajusta dando mayor importancia a las observaciones del conjunto de datos que los modelos anteriores manejaron incorrectamente [[13](#_ENREF_13)]. Al igual que el bagging, el boosting puede utilizarse tanto para problemas de regresión como de clasificación. Entre los algoritmos de boosting más conocidos se encuentran [[13](#_ENREF_13)]:

1. **Adaptive Boosting (AdaBoost)**: Este algoritmo ajusta los pesos de las observaciones en cada iteración, enfocándose en las instancias mal clasificadas por los modelos anteriores.

2. **Stochastic Gradient Boosting (SGB)**: Desarrollado por Friedman (2001), este método utiliza gradientes para minimizar la función de pérdida en cada iteración.

3. **Extreme Gradient Boosting (XGBoost)**: También conocido como XGB, fue propuesto por Friedman en el 2000 y es una implementación optimizada y escalable de gradient boosting.

## Métodos de evaluación.

Para entrenar y evaluar modelos de ABSA, es fundamental contar con conjunto de datos cuidadosamente etiquetados. Estos conjunto de datos sirven como referencia para comparar el rendimiento de diferentes modelos y permitir una evaluación precisa de su desempeño en tareas como la extracción de aspectos y la clasificación del sentimiento asociado a cada uno. Los conjunto de datos utilizados en ABSA suelen estar diseñados para abordar diferentes subtareas. Por ejemplo, para la extracción de aspectos, los conjunto de datos se etiquetan manualmente con los aspectos relevantes extraídos de las reseñas. En cambio, para la subtarea de clasificación de sentimiento por aspecto, los conjunto de datos se etiquetan con la polaridad (positiva, negativa o neutral) asociada a cada aspecto. Es importante destacar que la calidad de los conjunto de datos utilizados en ABSA juega un papel crucial en la calidad de los modelos entrenados. Un conjunto de datos bien diseñado, con anotaciones precisas y un tamaño adecuado, permitirá obtener resultados más robustos y confiables.

### Conjunto de datos.

El análisis de sentimientos basado en aspectos se apoya en conjuntos de datos anotados para su desarrollo. Entre los más utilizados [[20](#_ENREF_20),[21](#_ENREF_21), [30,32](#_ENREF_30)] se encuentran los proporcionados por SemEval 2014 y 2016. SemEval, una serie de talleres internacionales de investigación de procesamiento de lenguaje natural, promueve el avance en el análisis semántico y la creación de conjuntos de datos anotados de alta calidad. De los conjunto de datos de 2014 y 2016, los más utilizados son los relacionados con hoteles, restaurantes y laptops, disponibles en varios idiomas. Estos conjuntos de datos están etiquetados con la estructura E#A, donde E representa la entidad (restaurante, servicio, comida) y A el aspecto (precio, calidad) al que se refiere la opinión. Cada etiqueta incluye la polaridad asociada a la entidad y el aspecto [[55](#_ENREF_55)].

Los conjunto de datos encontrados en la literatura para la tarea de Análisis de Sentimientos Basado en Aspecto para el idioma español son:

* TASS 2014 SocialTV: Este conjunto de datos contiene aproximadamente 2773 tuits en español, centrados en la final de la Copa del Rey en España. A pesar de estar en español, las etiquetas abarcan las tareas de detección de la categoría del aspecto y clasificación del sentimiento del aspecto [[31](#_ENREF_31)].
* Semeval 2016 Restaurant Reviews Spanish Corpus: Está compuesto por reseñas etiquetadas con su polaridad (positiva, negativa o neutral) y con los aspectos a los que se refieren (comida, servicio, precio, ambiente, entre otros) en el contexto de la tarea 5 de SemEval-2016 [[55](#_ENREF_55)].

### Métricas utilizadas

**Las métricas son valores numéricos que miden el rendimiento de un modelo de aprendizaje automático cuando se trata de asignar observaciones a ciertas clases.** Generalmente, el rendimiento se presenta en un rango de 0 a 1, donde la puntuación 1 corresponde a un modelo perfecto, clasificado correctamente [[17](#_ENREF_17)].

En la actualidad existen un gran número de métricas, pero a continuación solo se describen las métricas utilizadas en este trabajo las cuales fueron una recopilación de las más usadas en las bibliografías.

Para entender el correcto funcionamiento de cada una de las métricas es fundamental conocer los siguientes 4 conceptos [[44](#_ENREF_44)]:

**Verdaderos Positivos (VP):** Se refieren cuando la clasificación es positiva y la predicción realizada también lo es.

**Verdaderos Negativos (VN):** Cuando la clasificación es negativa y el modelo también predice negativa.

**Falsos Positivos (FP):** Se refiere cuando el modelo realiza una predicción de positivo, cuando realmente era negativo, es decir el modelo se equivoca.

**Falsos Negativos (FN):** Cuando el modelo predice negativo cuando realmente es positivo, en este caso también se equivoca.

### Precisión

Es la razón entre el número de documentos clasificados correctamente como pertenecientes a la clase A y el número total de documentos de que han sido clasificados por el modelo como de clase A. La precisión mide la proporción de identificaciones positivas que son realmente correctas [[44](#_ENREF_44), [56](#_ENREF_56)].

**Precisión = 𝑇𝑃 / 𝑇𝑃 + 𝐹𝑃**

### Recall

La métrica Recall (Exhaustividad) es la relación entre los documentos clasificados correctamente como pertenecientes la clase A y la suma de todos los documentos de la clase A [[44](#_ENREF_44), [56](#_ENREF_56)].

**Recall = 𝑇𝑃 / 𝑇𝑃 + 𝐹𝑁**

### F1-Score

El valor F1 se utiliza para combinar las medidas de precisión y recall en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones [[44](#_ENREF_44), [56](#_ENREF_56)].

F1 se calcula haciendo la media armónica entre la precisión y la exhaustividad:

**𝐹1 = 2 ∗ (𝑝𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 ∗ 𝑟𝑒𝑐𝑎𝑙𝑙 /𝑝𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 + 𝑟𝑒𝑐𝑎𝑙l)**

### Accuracy

La exactitud (Accuracy) mide el porcentaje de casos que el modelo ha acertado. Esta es una de las métricas más utilizadas. El problema con la exactitud es que nos puede llevar al engaño, es decir, puede hacer que un modelo malo parezca que es mucho mejor de lo que es. Esto se debe a que el Accuracy valora el rendimiento del modelo teniendo en cuenta todas las clases del sistema y si estas están desbalanceadas los valores no van a ser confiables [[44](#_ENREF_44), [56](#_ENREF_56)].

**Accuracy = (𝑇𝑃 + 𝑇𝑁) / (𝑇𝑃 + 𝑇𝑁 + 𝐹𝑃 + 𝐹𝑁)**

## Conclusiones parciales.

Luego del análisis de lo planteado en este capítulo se llegaron a las siguientes conclusiones:

* Ha existido en los últimos años un aumento por parte de los investigadores en el área del Análisis de Sentimientos basado en Aspectos.
* Realizar la extracción de aspectos de las opiniones en español sigue siendo un gran desafío, ya que la mayoría de las soluciones que reportan buena tasa de eficiencia se encuentra en inglés.
* Se aprecia que de los métodos utilizados para el análisis de sentimientos el más utilizado es el método supervisado ya que ha sido el que ha mostrado mejores resultados.
* Las soluciones reportadas en la literatura, que emplean aprendizaje supervisado, precisan la existencia de suficientes datos para entrenamiento, los cuales existen muy pocos para idiomas como el español.
* Se analizaron los modelos de *Deep Learning* que utilizan la estructura *Transformers*, estos moldeos pre-entrenados son los que ofrecen mejores resultados en la actualidad*.*
* Se describieron las métricas más utilizadas en la bibliografía para valorar la efectividad y la calidad de los algoritmos automáticos.
* Se observaron los aspectos generales de la investigación y su marco teórico.

# Capítulo 2: Solución de análisis de sentimientos basado en aspectos para idioma español.

Este capítulo presenta una descripción detallada de la solución propuesta para el análisis de sentimientos basado en aspectos, enfocada específicamente en el idioma español. Se describen los principales conceptos y entidades que conforman la solución, junto con un esquema general que ilustra su diseño e implementación. Además, se explica paso a paso el funcionamiento del método propuesto, brindando una comprensión completa de su lógica y funcionamiento interno

## Descripción de la solución.

El método propuesto tiene como objetivo extraer los aspectos de las opiniones y determinar si estos aspectos expresan un sentimiento positivo o negativo utilizando un enfoque de aprendizaje supervisado. Este método está comprendido por tres pasos principales: (1) Preprocesamiento, (2) Extracción de aspectos y (3) Predicción de polaridad.

El algoritmo presentado fue desarrollado en el lenguaje de programación Python, usando las bibliotecas de PLN. En la figura 2 se muestra el flujo general de trabajo, donde se especifican los elementos principales de cada paso.

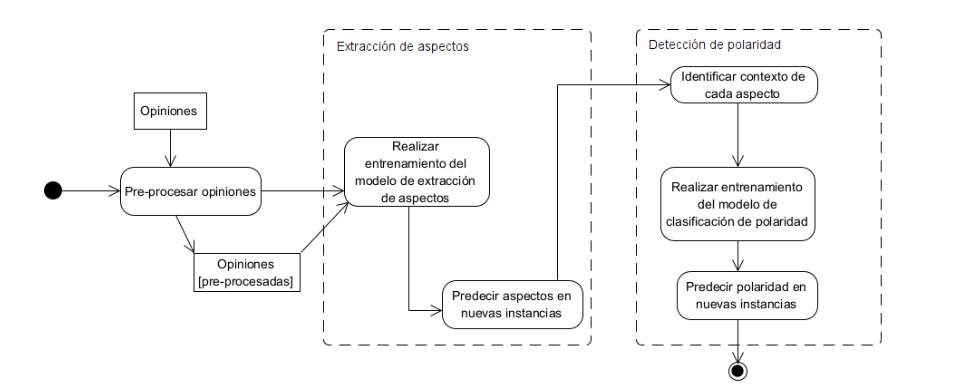


Figura 2. Esquema general del flujo del ABSA utilizando un enfoque supervisado.

### Preprocesamiento

El análisis de sentimientos basado en aspectos requiere un preprocesamiento específico para preparar los datos de texto y facilitar la extracción de información relevante en las opiniones. Si bien las tareas de preprocesamiento varían dependiendo del objetivo, en el caso del entrenamiento de los modelos *Transformers*, ciertas etapas son cruciales e indispensables, mientras que otras pueden omitirse.

Los modelos *Transformers*, han sido entrenados con millones de texto que conserva información como las mayúsculas y la puntuación. Por lo tanto, no es necesario convertir el texto a minúsculas ni eliminar los signos de puntuación, ya que el modelo ha aprendido a utilizar estas características para generar representaciones de alta calidad.

De igual manera, la eliminación de *stopwords* (palabras vacías como artículos y preposiciones) no es necesaria, dado que el corpus de entrenamiento ya lo incluye y su *tokenizador* las maneja sin problemas. En el contexto de la extracción de aspectos, una palabra como "servicio" puede ser un aspecto, pero también puede ser parte de una frase como "calidad de la comida". Eliminar las stopwords en estos casos podría implicar la pérdida de información relevante.

El primer paso en el preprocesamiento fue eliminar los caracteres especiales, dado que estos no son entendidos por los modelos, luego para la fase de entrenamiento del modelo fue necesario llevar el texto a un formato específico que fuese entendido por el modelo de lenguaje. Los datos de entrenamiento están compuestos por tres columnas, una con las opiniones, otra con el aspecto sobre el cual habla la opinión y una tercera que indica la polaridad del aspecto en dicha opinión.

En la Figura 3 se muestra como quedaría la representación para algunas reseñas. Por ejemplo, para la primera reseña: “La comida estuvo sabrosa”, el aspecto relevante es “comida” por tanto, en la columna objetivo se asigna un 1 en la posición de la palabra que constituye un aspecto a predecir y 0 en el resto de las posiciones ya que no son relevantes para la tarea. En el caso de la clasificación de polaridad la columna objetivo es un valor numérico que mapea cada una de los aspectos de la oración: no es aspecto (-1), negativo (0) o positivo (1).

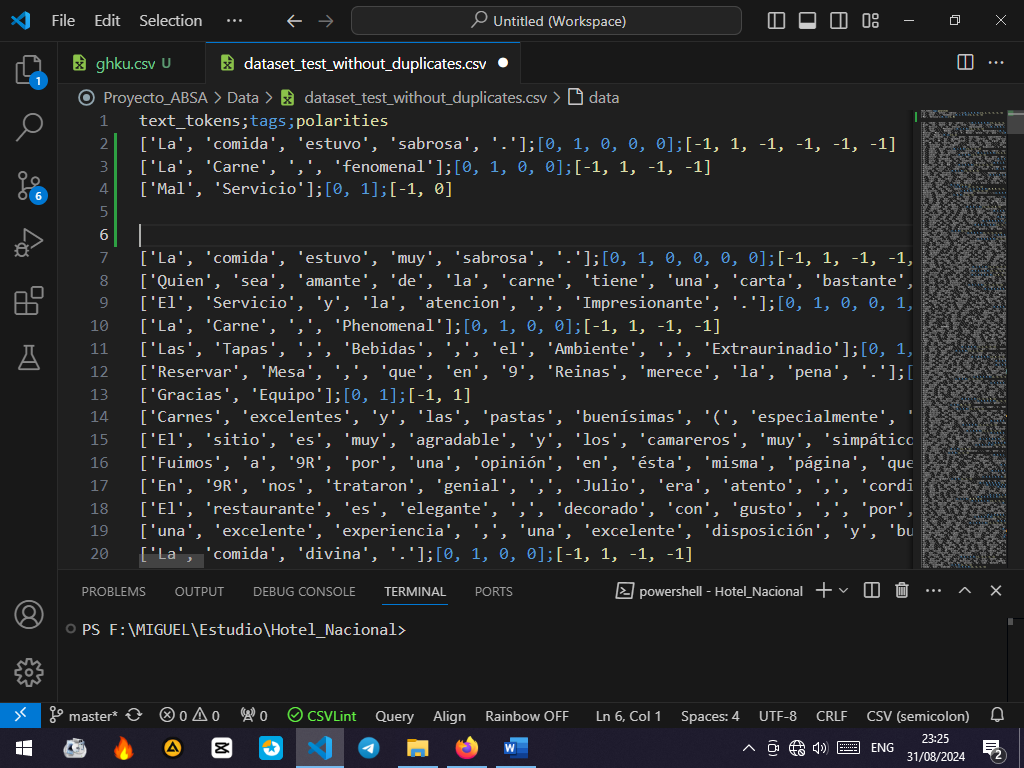


Figura 3. Representación de las reseñas.

### Extracción de aspectos.

La extracción de aspectos se abordó mediante la aplicación de una técnica de ensamble, un enfoque que integra múltiples modelos base en un mismo marco para obtener un modelo de clasificación más robusto y preciso. La efectividad de este método depende de varios factores, entre los que se incluyen la forma en que se entrenan los modelos base y cómo se combinan sus predicciones [[49](#_ENREF_13)]. En este trabajo, se propone un enfoque basado en ensamble por votación promedio, que combina los modelos base BETO, BERT, ALBERT-base, ALBERT-large y ALBERT-xx-large.

El proceso de extracción de aspectos se estructura en tres fases principales. En la primera fase, cada modelo base es sometido a un proceso de ajuste fino (*fine-tunning*), utilizando el corpus de reseñas de SemEval 2016, adaptando los parámetros pre-entrenados a la tarea específica de extracción de aspectos. Durante este proceso, se configura una capa de clasificación final para generar probabilidades binarias (0-1) correspondientes a las etiquetas de Aspecto y No Aspecto. Esta fase es crucial, ya que permite que cada modelo base se especialice en reconocer los patrones lingüísticos asociados a los aspectos en el contexto del idioma español. Para lograr esto, se añadió una capa lineal (Linear) completamente conectada a la salida de cada modelo, esta toma la salida de la última capa de la arquitectura del modelo y la proyecta en un espacio de características de tamaño igual al número de clases que se desea clasificar, en este caso, dos dimensiones (aspecto o no aspecto).

Durante el entrenamiento, se utilizó el algoritmo de optimización *Adam*, un método de descenso de gradiente estocástico (*Stochastic Gradient Descent, SGD)* que ajusta la tasa de aprendizaje para cada parámetro individual en función de los momentos de primer y segundo orden del gradiente [[60](#_ENREF_13)]. Este enfoque permite una convergencia más rápida y estable en problemas de optimización, como es el caso del entrenamiento de redes neuronales. La tasa de aprendizaje inicial se definió en 2e-5, un valor comúnmente utilizado en tareas de procesamiento de lenguaje natural. Además, se empleó la entropía cruzada (*Cross Entropy*) como función de pérdida, una medida que cuantifica la diferencia entre las predicciones del modelo y las etiquetas reales de los datos [[61](#_ENREF_13)]. Esta función es ampliamente utilizada en problemas de clasificación, ya que permite minimizar la discrepancia entre las distribuciones de probabilidad predichas y las verdaderas.

En la segunda fase, los modelos calibrados procesan el conjunto de prueba del mismo corpus, produciendo para cada token un vector de probabilidades que refleja su asociación con la clase objetivo. Este paso es esencial para evaluar el rendimiento de los modelos y asegurar que las predicciones sean consistentes y precisas. Cada modelo genera una salida independiente, lo que permite capturar diferentes perspectivas y matices en la identificación de aspectos.

Finalmente, en la tercera fase, las salidas de los cinco modelos se fusionan mediante un promedio aritmético. Esta operación genera un vector consolidado donde cada valor representa la probabilidad de consenso de que un token constituya un aspecto. Este enfoque de fusión permite combinar las fortalezas de cada modelo, reduciendo el impacto de posibles errores individuales y aumentando la confiabilidad de las predicciones. La figura 4 ilustra un ejemplo de extracción de aspectos de la reseña: “El servicio muy justo, el ambiente un poco decadente y la paella sin sabor”. En este ejemplo, el promedio de las predicciones del modelo da como resultado una probabilidad alta para los tokens “servicio”, “ambiente” y “paella”, identificándolos como aspectos.

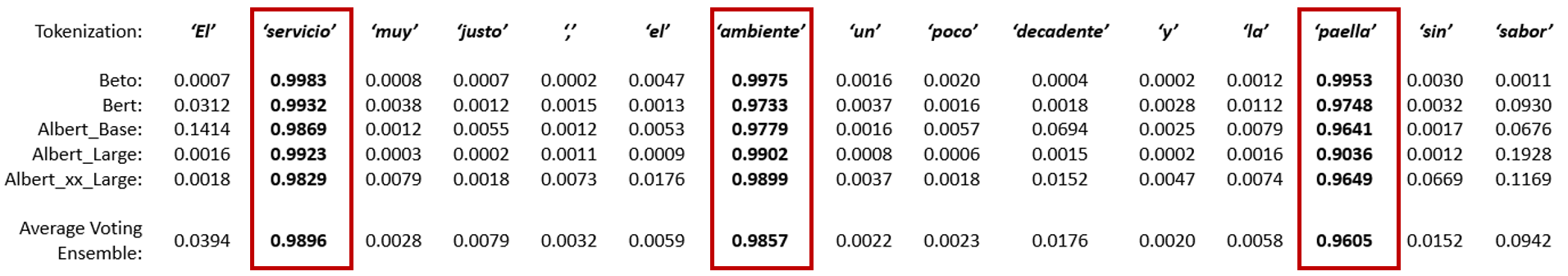


Figura 4. Ejemplo de identificación de aspectos con el uso del método de ensamble de votación promedio.

### Determinación de la polaridad.

#### Selección de Características Sentimentales

Una vez identificados los aspectos mencionados en el texto, el siguiente paso es determinar la **polaridad** de la opinión expresada sobre cada uno de ellos. La polaridad se refiere a la orientación del sentimiento, es decir, si la opinión es positiva o negativa respecto al aspecto en cuestión. Para lograr esto, es crucial seleccionar las **características o palabras clave** que expresan el sentimiento hacia un aspecto específico. Estas características son las que permiten al modelo inferir la intención del autor y clasificar adecuadamente la polaridad. En este trabajo, se utiliza un **método híbrido** para la selección de características sentimentales, que combina dos técnicas principales: la **ventana de palabras** y el **análisis de dependencias sintácticas**. Este enfoque híbrido permite aprovechar las fortalezas de ambas técnicas, compensando las limitaciones de una con las ventajas de la otra, lo que resulta en un método más robusto y preciso.

**Técnica de la ventana de palabras**

La técnica de la ventana de palabras consiste en definir un rango de palabras alrededor del aspecto identificado, que se considera como el contexto inmediato donde se expresa el sentimiento. Por ejemplo, si el aspecto es "servicio" en la oración "El servicio fue excelente, pero el ambiente dejó mucho que desear", la ventana de palabras podría incluir las palabras "fue excelente". Este enfoque se basa en la suposición de que las palabras cercanas al aspecto son las que contienen la información más relevante sobre su polaridad. Sin embargo, esta técnica tiene limitaciones, ya que no siempre captura relaciones más complejas o dependencias sintácticas que pueden estar más alejadas en la oración.

**Uso de dependencias sintácticas**

El análisis de dependencias sintácticas, por otro lado, se enfoca en identificar las relaciones gramaticales entre las palabras de una oración. Esta técnica permite capturar conexiones más profundas y contextuales entre el aspecto y las palabras que expresan el sentimiento, incluso si estas no están físicamente cerca en el texto. Por ejemplo, en la oración "El servicio nocturno en el restaurante del hotel fue terrible", la palabra "terrible" está sintácticamente relacionada con "servicio", a pesar de no estar inmediatamente adyacente. Este método es especialmente útil para capturar sentimientos expresados de manera indirecta o en oraciones con estructuras complejas.

#### Clasificación de sentimientos

Una vez seleccionadas las palabras que expresan la opinión sobre un aspecto, el siguiente paso es clasificar dicha opinión en positiva o negativa. Para lograr esto, se empleó un enfoque basado en modelos de lenguaje pre-entrenados, similar al utilizado para la extracción de aspectos. En este caso, se seleccionó el modelo **BERTIN\_Large** [[62](#_ENREF_21)], una versión de **RoBERTa** específicamente adaptada para el idioma español. BERTIN\_Large cuenta con una arquitectura robusta, compuesta por 24 capas, 16 cabezas de atención por capa, un tamaño oculto de 1024 dimensiones y un total de 355 millones de parámetros. Para adaptar BERTIN\_Large a la tarea específica de clasificación de polaridad, se realizó un **ajuste fino** utilizando un conjunto de datos de entrenamiento compuesto por fragmentos de texto que incluyen el aspecto junto con su contexto (la ventana de palabras y las dependencias sintácticas asociadas). Cada fragmento de texto está acompañado de una etiqueta numérica que indica la polarid**ad, 0** para opiniones negativ**as y 1** para opiniones. Este proceso de ajuste fino permite que el modelo aprenda a identificar patrones específicos en el texto que están relacionados con la expresión de sentimientos hacia un aspecto determinado.

La arquitectura del modelo de clasificación se basa en BERTIN\_Large, pero se le añadió **capa lineal** y **función de activación Softmax** para la asignación de clases. La capa lineal toma la salida de BERTIN\_Large, que es un vector de 1024 dimensiones, y lo proyecta en un espacio de 2 dimensiones, donde cada dimensión corresponde a una de las clases de polaridad. La función Softmax se utiliza para normalizar las salidas de la capa lineal, convirtiéndolas en probabilidades que indican la confianza del modelo en cada una de las clases. Para entrenar el modelo, se utilizó la **función de pérdida de Entropía y c**omo algoritmo de optimización, se empleo Adam

## Desarrollo de la solución de análisis de sentimientos basado en aspectos.

En esta sección, se detallan los elementos clave de la ingeniería de software utilizados en la construcción del sistema. Entre estos elementos se encuentra el modelo de dominio, el diagrama de casos de uso del sistema, la descripción detallada de los casos de uso, los requisitos funcionales y no funcionales y la descripción de cada uno de los paquetes y clases del software.

### Modelo de dominio.

En este capítulo se revisa el modelo de dominio, el cual es una representación visual de clases conceptuales u objetos de situaciones reales en un dominio. Los modelos de dominio también se han denominado modelos conceptuales o modelos de objetos de dominio. Los modelos de dominio son la técnica más importante en las actividades de análisis. Es fundamental entender que solo cubren conceptos que son comprensibles para el cliente y el usuario [[57](#_ENREF_57)]. Aplicando la notación UML se ilustra un modelo de dominio en la figura 4.

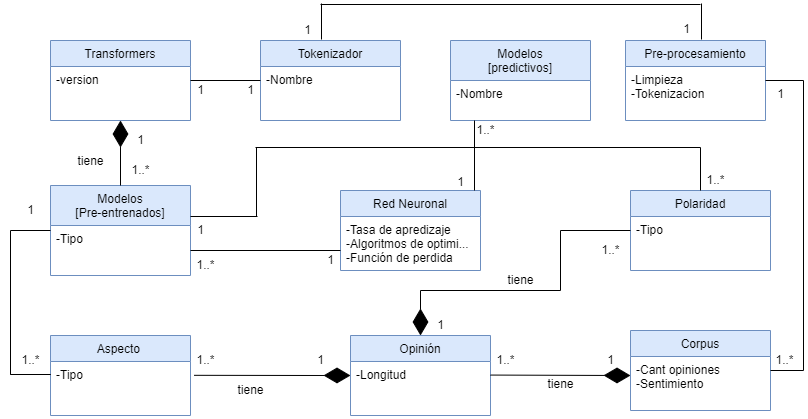


Figura 5. Modelo de dominio.

### Glosario de términos.

A continuación, en la , se muestra el glosario de términos que facilita tener una comprensión más detallada de los conceptos empleados en el modelo de dominio.

Tabla 3. Glosario de término.

|  |  |
| --- | --- |
| **Nombre del concepto** | **Descripción** |
| Opinión | Texto que expresa o manifiesta el punto de vista de una persona hacia un elemento determinado |
| Corpus | Colección de textos, en este caso, de opiniones. También conocido como conjunto de datos |
| Polaridad | Representa el sentimiento hacia el aspecto, que puede ser negativo, positivo o neutro. |
| Aspecto | Representa un atributo o característica de una determinada entidad |
| Modelos pre-entrenados | Modelos entrenados con volúmenes masivos de datos, como BERT, RoBERTa, entre otros |
| Tokenizador | Función que se encarga de convertir el texto en tokens. |
| Red Neuronal | Arquitectura compuesta por distintos elementos utilizada para determinar aspectos y polaridad. |
| Preprocesamiento | Acciones de limpieza y organización de datos previo al entrenamiento y la predicción. |
| Transformers | Biblioteca que permite trabajar con distintos modelos que utilicen esta arquitectura. |
| Modelos predictivos | Entrenamiento resultante de la red neuronal. |

### Captura de requisitos.

### Requisitos funcionales.

R1: El sistema debe ser capaz entrenar un modelo de Deep Learning para extracción de aspectos.

R2: El sistema debe ser capaz de, dado un conjunto de opiniones, identificar y devolver los aspectos que los usuarios mencionan en sus comentarios.

R3: El sistema debe ser capaz de identificar la polaridad del sentimiento expresados en el texto relacionado con cada aspecto identificado.

R4: El sistema debe ser capaz de realizar limpieza del texto y correcciones gramaticales.

R5: El sistema debe permitir, dado un archivo en formato .csv o .txt, cargar los datos en memoria interna para procesarlos.

R6: El sistema debe permitir seleccionar una ruta donde se encuentra el archivo con opiniones que desea cargar el usuario.

R7: El sistema debe ser capaz de devolver el por ciento de opiniones positivas y negativas y los aspectos más mencionados dado un conjunto de opiniones.

R8: El sistema debe evaluar la calidad de la extracción de aspectos.

R9: El sistema debe evaluar la calidad de la clasificación de sentimientos.

R10: El sistema debe ser capaz de entrenar un modelo para clasificar polaridad en opiniones.

R11: El sistema debe ser capaz de crear los vectores de palabras para introducir a la red neuronal.

R12: El sistema de mostrar los resultados en forma de tablas y gráficos.

### Requisitos no funcionales.

R1: El sistema debe demostrar una alta precisión en la clasificación de aspectos.

R2: El sistema debe demostrar una alta precisión en la clasificación de la polaridad.

R3: El sistema debe ser capaz de escalar en forma horizontal.

R4: El método debe ser capaz de generalizar eficientemente a datos no vistos.

R5: El sistema debe ser capaz de adaptarse y rendir en dominios diferentes.

R6: El método debe demostrar resistencia ante ruido en los datos y ambigüedad lingüística.

R7: El sistema debe ser fácil de mantener y actualizar.

R8: El sistema debe ser fácil de usar y comprender para los usuarios.

### Diagrama de caso de uso del sistema.

El diagrama de casos de uso del sistema ofrece una visión general de las funcionalidades principales que el sistema debe proporcionar, permitiendo comprender las interacciones entre los usuarios y el sistema desde un punto de vista estático.

La Figura 5 presenta el diagrama de casos de uso desarrollado para la presente propuesta, el cual ilustra las diferentes acciones que el sistema permite realizar y las relaciones entre los distintos actores involucrados.

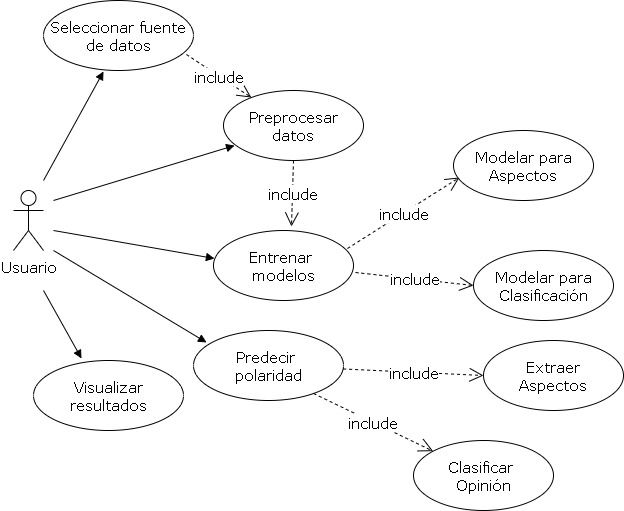


Figura 6.Diagrama de casos de uso.

### Descripción detallada de los casos de uso.

A continuación, se presenta la descripción detallada para cada caso de uso:

Tabla 4. Descripción del CUS “Seleccionar fuente de datos”.

|  |  |
| --- | --- |
| **Caso de uso** | Seleccionar fuente de datos |
| **Actores** | Usuario (inicia) |
| **Descripción** | EL caso de uso comienza cuando el usuario inicia la aplicación y busca en los archivos locales un conjunto de datos en formato .csv donde tiene las opiniones almacenadas. |
| **Requisitos funcionales** | R5, R6 |
| **Casos de uso asociados** | --- |
| **Precondición** | Debe existir un conjunto de datos local en formato csv que contenga las opiniones. |
| **Postcondición** | Se cargan las opiniones para comenzar el análisis. |
| **Requisitos no funcionales** | R7, R8 |

Tabla 5. Descripción del CUS “Preprocesar datos”.

|  |  |
| --- | --- |
| **Caso de uso** | Preprocesar datos |
| **Actores** | Usuario (inicia) |
| **Descripción** | El caso de uso comienza cuando se han cargado en memoria interna las opiniones que se desean analizar, se eliminan los caracteres especiales. Además, para la fase de entrenamiento en el preprocesamiento los datos se llevan a un formato que sea comprendido por la red neuronal, es decir, se construye un vector binario, en el caso del entrenamiento para extracción de aspectos y en el caso de clasificación de polaridad se construye la ventana de palabras, se transforma la columna clase en valores numéricos, se tokenizan las reseñas, entre otros. |
| **Requisitos funcionales** | R4, R11 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las opiniones a analizar. |
| **Postcondición** | Se devuelven las opiniones preprocesadas. |
| **Requisitos no funcionales** | R3, R5 |

Tabla 6. Descripción del CUS “Entrenar modelos”.

|  |  |
| --- | --- |
| **Caso de uso** | Entrenar modelos |
| **Actores** | Usuario (inicia) |
| **Descripción** | El caso de uso comienza cuando se han seleccionado un conjunto de datos y este ha sido pre-procesado para la tarea que se desea realizar. Luego se escoge si se va a realizar entrenamiento de un modelo de extracción de aspectos o de clasificación de polaridad y se comienza con el entrenamiento. El caso de uso termina cuando se ha obtenido el modelo entrenado para la tarea en cuestión. |
| **Requisitos funcionales** | R1, R10 |
| **Casos de uso asociados** | Modelar para aspectos <include>  Modelar para clasificación <include> |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las reseñas a analizar preprocesadas. |
| **Postcondición** | Se obtienen el modelo entrenado para la tarea seleccionada. |
| **Requisitos no funcionales** | R3, R4, R5 |

Tabla 7. Descripción del CUS “Modelar para aspectos”.

|  |  |
| --- | --- |
| **Caso de uso** | Modelar para aspectos |
| **Actores** | Usuario |
| **Descripción** | El caso de uso comienza cuando todas las reseñas han sido preprocesadas, luego comienza el proceso de entrenamiento, donde se inicializa el modelo de Deep Learning pre-entrenado combinado con la capa lineal y la función de activación, se van seleccionando lotes de datos que son pasados al modelo, se estima el error del modelo con la función de pérdida *Cross Entropy*, se propaga el error hacia atrás por la red y se vuelven a ajustar los pesos. El caso de uso termina cuando se ha entrenado el modelo todas las iteraciones que se definieron inicialmente. |
| **Requisitos funcionales** | R1, R8 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las reseñas preprocesadas con la columna clase con la estructura de vector binario utilizada para denotar la presencia o no de aspectos. |
| **Postcondición** | Se obtienen un modelo entrenado para la tarea de extracción de aspectos que puede ser utilizado posteriormente para realizar predicciones. |
| **Requisitos no funcionales** | R1, R3 |

Tabla 8. Descripción del CUS “Modelar para clasificación”.

|  |  |
| --- | --- |
| **Caso de uso** | Modelar para clasificación |
| **Actores** | Usuario |
| **Descripción** | El caso de uso comienza cuando todas las opiniones han sido preprocesadas, luego comienza el proceso de entrenamiento, donde se inicializa el modelo de Deep Learning pre-entrenado combinado con la capa lineal y la función de activación, se van seleccionando lotes de datos que son pasados al modelo, se estima el error del modelo con la función de pérdida *Cross Entropy*, se propaga el error hacia atrás por la red y se vuelven a ajustar los pesos. El caso de uso termina cuando se ha entrenado el modelo todas las iteraciones que se definieron inicialmente. |
| **Requisitos funcionales** | R9, R10 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las opiniones preprocesadas con la columna clase que contiene las etiquetas positivas o negativo. |
| **Postcondición** | Se obtienen un modelo entrenado para la tarea de clasificación de polaridad que puede ser utilizado posteriormente para realizar predicciones. |
| **Requisitos no funcionales** | R2, R3, R4 |

Tabla 9. Descripción del CUS “Predecir polaridad”.

|  |  |
| --- | --- |
| **Caso de uso** | Predecir polaridad |
| **Actores** | Usuario (inicia) |
| **Descripción** | El caso de uso comienza cuando se carga en el sistema un conjunto de datos, luego estos datos son preprocesados y una vez que están listos son pasados al módulo de extracción de aspectos donde se seleccionan los aspectos sobre los cuales se emite una opinión y se clasifica su polaridad. El caso de uso termina cuando se han extraído los aspectos de cada opinión y se ha clasificado la polaridad de cada opinión para cada uno de los aspectos. |
| **Requisitos funcionales** | R2, R3, R7 |
| **Casos de uso asociados** | Extraer aspectos <include>  Clasificar opinión <include> |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las reseñas a analizar. |
| **Postcondición** | Se obtienen por cada aspecto sus reseñas asociadas clasificadas en positivas o negativas. |
| **Requisitos no funcionales** | R1, R2, R6, R7 |

Tabla 10. Descripción del CUS “Extraer aspectos”.

|  |  |
| --- | --- |
| **Caso de uso** | Extraer aspectos |
| **Actores** | Usuario |
| **Descripción** | El caso de uso comienza cuando todas las reseñas han sido preprocesadas, luego estas pasan al proceso de extracción de aspectos para ello se carga el modelo de Deep Learning pre-entrenado y ajustado al dominio específico y se hace la predicción de los aspectos que aparecen en cada una de las opiniones. |
| **Requisitos funcionales** | R2 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las reseñas a analizar preprocesadas y se debe tener el modelo ajustado al dominio. |
| **Postcondición** | Se obtienen las reseñas y el o los aspectos que se encontraron en cada una de ellas. |
| **Requisitos no funcionales** | R1, R7 |

Tabla 11. Descripción del CUS “Clasificar opinión”.

|  |  |
| --- | --- |
| **Caso de uso** | Clasificar opinión |
| **Actores** | Usuario |
| **Descripción** | El caso de uso comienza cuando se han se han extraído todos los aspectos encontrados en las reseñas, luego se localizan las palabras de sentimiento asociadas a cada aspecto en los comentarios y se clasifica la polaridad para cada aspecto creando una ventana de palabras y utilizando el entrenamiento almacenado en el modelo de Deep Learning. |
| **Requisitos funcionales** | R3, R7 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se debe suministrar un conjunto de datos que contenga las reseñas a analizar preprocesadas. |
| **Postcondición** | Se obtienen las reseñas por cada aspecto clasificadas en positivas o negativas. |
| **Requisitos no funcionales** | R2, R6 |

Tabla 12.Descripción del CUS “Visualizar resultados”.

|  |  |
| --- | --- |
| **Caso de uso** | Visualizar resultados |
| **Actores** | Usuario (inicia) |
| **Descripción** | El caso de uso comienza cuando se han realizados las predicciones a cada una de las opiniones y el usuario desea mostrar los resultados a través de tablas y graficas. |
| **Requisitos funcionales** | R12 |
| **Casos de uso asociados** | --- |
| **Precondición** | Se deben tener cada una de las opiniones clasificadas según su aspecto en positivas y negativas. |
| **Postcondición** | Se obtienen los resultados en forma de tablas y gráficos. |
| **Requisitos no funcionales** | R7, R8 |

### Descripción de paquetes.

A continuación, en la , se detallan los paquetes utilizados en el proyecto.

Tabla 13. Descripción de paquetes

|  |  |
| --- | --- |
| Paquete | Descripción |
| Extracción de aspectos: | En este paquete se encuentran las funciones que permiten realizar el entrenamiento de un modelo de *Transformers* para la tarea de extracción de aspectos, así como los métodos que permiten predecir los aspectos en nuevas reseñas. |
| Data | Paquete destinado al almacenamiento del corpus sin procesar y después de ser procesado |
| Preprocesamiento: | Paquete que almacena las clases necesarias para la tarea del preprocesamiento. Algunas de esas funcionalidades fueron eliminar los caracteres especiales y llevar los datos de entrenamiento a un formato que fuese comprendido por la red. |
| Model | Paquete que almacena los modelos que fueron utilizados para entrenar, así como los modelos resultantes del entrenamiento |
| Determinar polaridad: | Este paquete contiene métodos que permiten entrenar el modelo de polaridad, así como almacenar su entrenamiento, definir sus hiperparámetros, entre otros. |
| Evaluación: | Este paquete contiene los métodos que permiten medir la calidad de la solución propuesta, así como funciones que grafican los resultados obtenidos. |

### Descripción de clases.

A continuación, en la , se detallan las clases utilizadas en el proyecto.

Tabla 14. Descripción de las clases utilizadas.

|  |  |
| --- | --- |
| Clases | Descripción |
| Main | Clase que integra todo el flujo del método propuesto, es la encargada de darle un orden lógico a todos los pasos de ejecución y dar la salida final del algoritmo. |
| ABSAModel | Clase abstracta utilizada para definir métodos comunes tanto para el modelo de polaridad como para el de extracción de aspectos |
| AspectTermExtraction | Se utiliza para realizar, tanto entrenamiento del modelo de extracción de aspectos como predicción en nuevas instancias. |
| Conjunto de datosLoaderAspect | Clase que se encarga de cargar y preparar los datos relacionados con los aspectos para su procesamiento y entrenamiento. |
| Conjunto de datosLoaderPolarity | Similar a *Conjunto de datosLoaderAspect*, pero se enfoca en cargar y preparar datos relacionados con la polaridad de los sentimientos |
| ABSAPipeline | Representa la tubería o flujo de procesamiento que combina las diferentes etapas del análisis de sentimiento basado en aspectos, desde la identificación de aspectos hasta la clasificación de sentimientos. |
| Pytorch.nn.Module | Parte del *framework PyTorch*, es una interfaz que se utiliza para definir modelos de aprendizaje profundo. |

## Problemas frecuentes.

En el campo del análisis de sentimientos, existen varios problemas frecuentes que pueden afectar la calidad del software y, en particular, la precisión del modelo entrenado. Estos problemas, si no se abordan adecuadamente, pueden llevar a resultados poco confiables y a un rendimiento deficiente del sistema. A continuación, se describen algunos de los problemas más comunes y las estrategias para mitigarlos.

### Insuficiencia de datos.

Una de las etapas más críticas en el análisis de sentimientos es el entrenamiento del modelo, el cual requiere un conjunto de datos previamente etiquetado. Sin embargo, uno de los problemas más frecuentes es la insuficiencia de datos o la baja calidad de los mismos. Si el conjunto de datos es pequeño o está mal etiquetado, el modelo aprenderá de manera incorrecta, lo que se traducirá en una baja precisión al predecir nuevas instancias. Este problema es particularmente difícil de validar, ya que, aunque se pueda comprobar que el conjunto de datos supere un umbral mínimo (por ejemplo, 2000 opiniones), no siempre es posible garantizar la calidad de las etiquetas o la representatividad de los datos. Por ello, es fundamental seleccionar fuentes de datos confiables y oficiales, como conjuntos de datos estandarizados (por ejemplo, SemEval), que hayan sido validados por la comunidad científica. Además, es recomendable realizar una revisión manual de una muestra de los datos para asegurar que las etiquetas sean consistentes y precisas.

### Desbalance de clases.

El desbalance de clases ocurre cuando una clase tiene significativamente más ejemplos que otra en el conjunto de datos. Por ejemplo, si en un conjunto de 3000 opiniones, 2000 son positivas y 1000 son negativas, el modelo podría tener dificultades para aprender a clasificar correctamente las opiniones negativas, ya que estaría sesgado hacia la clase mayoritaria (positiva). Esto puede llevar a que el modelo clasifique incorrectamente muchas opiniones negativas como positivas, reduciendo su capacidad para generalizar. Para mitigar este problema, es importante verificar el balance de clases después de seleccionar la fuente de datos.

### Sobreajuste (*Overfitting*).

El sobreajuste es un problema común en los modelos de Deep Learning, donde el modelo se ajusta demasiado a los datos de entrenamiento, perdiendo su capacidad para generalizar a datos nuevos y no vistos. Esto ocurre cuando el modelo memoriza los patrones específicos del conjunto de entrenamiento, incluyendo el ruido y los detalles irrelevantes, en lugar de aprender características generalizables. Para evitar el sobreajuste, se implementaron estrategias durante el entrenamiento del modelo. Se añadieron capas de dropout a la red neuronal, que desactivan aleatoriamente una fracción de las neuronas durante el entrenamiento. Esto evita que el modelo dependa demasiado de ciertas neuronas y fomenta la generalización. Además se ajustó la tasa de aprendizaje del modelo para evitar que este converja demasiado rápido hacia un mínimo local. Una tasa de aprendizaje adecuada permite que el modelo explore mejor el espacio de parámetros y evite el sobreajuste.

## Conclusiones parciales.

Luego del análisis de este capítulo se pueden abordar las siguientes conclusiones:

* Se describió un flujo de trabajo estructurado para el análisis de sentimientos basado en aspectos, compuesto por tres etapas clave: **preprocesamiento, extracción de aspectos** y **predicción de polaridad**. Este enfoque sistemático garantiza un proceso eficiente y preciso para analizar opiniones.
* En el modelo propuesto, la **opinión** es la entidad protagonista, ya que es la entrada principal del método y su análisis permite obtener la clasificación de los aspectos y su polaridad. Esto resalta la importancia de procesar las opiniones de manera efectiva para aplicaciones prácticas.
* Los **modelos Transformers**, como BERTIN, desempeñan un papel crucial en el desarrollo del ABSA. Su capacidad para capturar el contexto y las relaciones semánticas, junto con su adaptabilidad mediante fine-tuning, los convierte en herramientas poderosas para la extracción de aspectos y la clasificación de polaridad.
* Se detalló la estructura del proyecto, incluyendo sus componentes principales y casos de uso, lo que brinda una comprensión clara de la problemática y su solución. La integración de técnicas como el aprendizaje conjunto y el uso de dependencias sintácticas demuestra un enfoque innovador y robusto.
* Se abordaron desafíos comunes, como la insuficiencia de datos, el desbalance de clases y el sobreajuste, proponiendo soluciones prácticas como la selección de fuentes confiables, el balanceo de clases y técnicas de regularización. Esto asegura un modelo preciso y generalizable.

# Capítulo 3: Evaluación de la solución propuesta.

Este capítulo presenta la validación del trabajo de investigación. Se detalla el proceso de evaluación utilizando las métricas accuracy, precisión, recall y F1-score. Además se comparan los resultados obtenidos con los reportados en la literatura científica para el mismo conjunto de datos. Este experimento permite una evaluación objetiva del rendimiento de los modelos propuestos, asegurando su competitividad frente a enfoques existentes.

## Marco de Evaluación.

La evaluación de la propuesta se estructura en dos etapas. Primero, se evalúa la precisión de la extracción de aspectos, considerando la subtarea de extracción de términos de opinión (*Opinion Term Extraction*, *Task 5 slot 2* de SemEval 2016) [[28](#_ENREF_22)], en el contexto del análisis de sentimientos basado en aspectos. Luego se evalúa el rendimiento en la clasificación de la polaridad de los aspectos (*Aspect Sentiment Classification, Task 5 slot 3 de SemEval 2016*) [[28](#_ENREF_22)].

Para ambas tareas, se seleccionan diez modelos de lenguaje basados en arquitecturas *Transformer*, los cuales fueron: BETO (BERT para idioma español), BERT multilingüe (*bert-multilingual-cased*), ELECTRA dos versiones (versión para español), BERTIN en dos tamaños, GPT-2-spanish, y AlBERT en tres tamaños diferentes. A diferencia de BETO, que se entrena solo en el idioma español, BERT multilingüe se entrena en una gran cantidad de texto en diferentes idiomas, incluyendo español, chino, árabe, hindi, portugués, francés, entre otros.

Los modelos se entrenan durante cinco etapas con un tamaño de lote (*batch size)* de ocho. La elección del tamaño de lote estuvo motivada por varios factores, entre ellos está que los tamaños de lote grandes pueden llevar a fluctuaciones en las actualizaciones de los gradientes. Además, tamaños de lotes más grandes pueden conducir a una convergencia más rápida, pero existe el riesgo de que el modelo se ajuste demasiado a los datos de entrenamiento y no generalice bien en nuevos datos [22].

En la tarea de extracción de aspectos, después de finalizar el entrenamiento, se calculan las métricas de accuracy, precisión, recall y F1 para cada modelo. Se seleccionaron los cinco mejores modelos en la métrica de F1-score, ya que es la métrica más apropiada para medir la calidad en conjuntos de datos desequilibrados. Estos modelos se integraron en un ensamble por votación máxima (*Voting Ensemble*). Esta técnica combina las predicciones individuales, seleccionando la etiqueta más frecuente para cada instancia, reduciendo el sesgo de modelos individuales.

Para la clasificación de sentimientos, se utilizaran los mismos modelos y parámetros, explorando diferentes estrategias para identificar las palabras asociadas a la polaridad de los aspectos. Se evalúan ventanas de palabras de distintos tamaños para determinar el mejor contexto alrededor del aspecto. También se realiza un análisis de dependencias para identificar las relaciones gramaticales entre los aspectos y los términos de opinión, como adjetivos o verbos modales. Por último, se combina la estrategia de ventana de palabras con el análisis sintáctico para obtener un enfoque híbrido, que captura tanto el contexto local como las relaciones estructurales.

## Descripción de las fuentes de datos

La evaluación experimental de la solución emplea el conjunto de datos de reseñas españolas ofrecido en SemEval-2016 para la Tarea 5: Análisis de Sentimientos Basado en Aspectos [[28](#_ENREF_22)]. Esta colección está formada por reseñas de restaurantes, etiquetadas con los aspectos que se deben identificar en un proceso automático (por ejemplo: servicio, comida, ambiente, atención, entre otros). Existen muy pocos conjuntos de datos españoles para evaluar este tipo de soluciones [63]. Por tanto, el conjunto seleccionado es el más utilizado. Los resultados se calculan utilizando las métricas Accuracy, Precisión, Recall y F1-score. A continuación en la , se muestra una caracterización del conjunto de datos ABSA-SemEval-2016.

Tabla 15. Caracterización del corpus ABSA-SemEval-2016

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ABSA-SemEval-2016 | Opiniones | Entrenamiento | Test | Aspectos |
| 2.951 | 2070 | 881 | 247 |

## Resultados y discurso.

A continuación, se muestran los resultados obtenidos en la extracción de aspecto y la clasificación de la polaridad para opiniones de idioma español, usando como conjunto de datos de SemEval 2016.

### Resultados de la evaluación de la extracción de aspectos.

En la , se muestra la configuración de los distintos hiperparámetros definidos para la ejecución de los experimentos con cada uno de los modelos utilizados.

Tabla 16. Configuración de los hiperparámetros de los modelos.

|  |  |
| --- | --- |
| **Hiperparámetros** | **Valor** |
| Semilla (*seed*) | 42 |
| Tasa de aprendizaje | 2e-5 |
| Cantidad de épocas (*epochs*) | 5 |
| Tamaño de lote | 8 |
| Tamaño máximo de la secuencia de palabras | 512 |

A continuación, en la , se muestran los resultados individuales de cada uno de los modelos y se señalan en negrita los mejores resultados.

Tabla 17. Evaluación de la calidad de la extracción de aspectos.

|  |  |  |  |
| --- | --- | --- | --- |
| Modelos | Precisión | Recall | F1 |
| Beto | **86.36** | 83.46 | 84.83 |
| Bert | 85.17 | 83.13 | 84.11 |
| Albert\_base | 82.38 | 83.00 | 82.69 |
| Albert\_large | 84.68 | 85.79 | **85.22** |
| Albert\_xx\_large | 82.33 | **86.27** | 84.14 |
| Bertin\_base | 78.31 | 83.12 | 80.44 |
| Bertin\_large | 84.03 | 81.45 | 82.67 |
| Electra\_small | 81.27 | 80.06 | 80.65 |
| Electra\_base | 82.85 | 82.29 | 82.57 |
| Gpt\_2 | 78.24 | 53.39 | 53.82 |

Estos resultados presentan diferencias significativas en el rendimiento de los modelos analizados. El modelo ALBERT-xx-large destaca con un recall del 86.27%, superando a todas las arquitecturas individuales. Esto indica la capacidad para identificar la mayoría de los aspectos presentes en el texto, minimizando falsos negativos. Sin embargo, su precisión de 82.33% y F1-score de 84.14% son ligeramente inferiores, lo que demuestra una tendencia a incluir términos irrelevantes como aspectos.

Por otra parte, los modelos como BETO y BERT multilingüe presentan equilibrio y adaptabilidad al español a pesar de las diferencias en sus arquitecturas. Además, se reafirma la eficacia de los modelos multilingües para tareas específicas en lenguas con recursos limitados.

Por último, es notable el bajo rendimiento del modelo GPT-2-spanish. Esto podría atribuirse a su naturaleza generativa no bidireccional, menos adecuada para tareas de etiquetado secuencial como la extracción de aspectos.

A continuación en la , se muestra la evaluación del ensamble por votación máxima (BETO + BERT + ALBERT-base + ALBERT-large + ALBERT-xx-large). Este ensamble obtiene el mejor resultado de precisión y F1-score, superando a cualquier modelo individual. Este resultado reafirma que la combinación de modelos con diferentes fortalezas mitiga sesgos individuales.

Tabla 18. Evaluación del ensamble por votación máxima

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ensamble | Modelos | Precisión | Recall | F1 |
| Votación máxima | Beto + Bert + Albert\_base + Albert\_large + Albert\_xx\_large | 87.09 | 85.84 | 86.45 |

### Análisis de los resultados de la extracción de aspectos.

En la , se muestra la comparación de la solución frente a otras soluciones reportadas en la literatura que emplean el mismo conjunto de datos y se evalúan para la misma subtarea en idioma español. La mayoría de las soluciones reportan sus resultados solo en términos de la métrica F1-score. Sin embargo, en esta investigación también se considera conveniente utilizar las demás métricas mencionadas al inicio de la sección, con el objetivo de tener una mejor comprensión del funcionamiento del modelo en dicha tarea.

Tabla 19. Comparación de la solución para la extracción de aspectos frente a otras soluciones reportadas en la literatura.

|  |  |  |  |
| --- | --- | --- | --- |
| **Soluciones** | **Precisión** | **Recall** | **F1** |
| GTI/C\* |  |  | 68.50% |
| GTI/U\* |  |  | 68.30% |
| IIT-T/U\* |  |  | 64.30% |
| TGB/C\* |  |  | 55.70% |
| SemEval-2016 – baseline\* |  |  | 51.80% |
| [[6](#_ENREF_39)] |  |  | 73,00% |
| [[22](#_ENREF_22)] | 76,70% | 81,50% | 76,70% |
| [[39](#_ENREF_39)] |  |  | 73,07% |
| [[40](#_ENREF_40)] |  |  | 65,40% |
| [[58](#_ENREF_40)] |  |  | 73.00% |
| [[59](#_ENREF_40)] |  |  | 71.70% |
| **BETO + BERT + ALBERT-base + ALBERT-large + ALBERT-xx-large** | **87.09%** | **85.84%** | **86.45%** |

\* Soluciones presentadas en SemEval-2016 Task 5 [[28](#_ENREF_28)]

Como se aprecia, la solución propuesta obtiene mejores resultados que todas las soluciones reportadas en la literatura para esta subtarea de extracción de aspectos en idioma español. Estos resultados corroboran la ventaja que supone el uso de los modelos de lenguaje adaptados a tareas específicas de procesamiento de lenguaje natural y el uso de técnicas de ensamble.

### Resultados de la clasificación de la polaridad.

Para la clasificación de la polaridad de los aspectos, se evalúan tres estrategias de selección de contexto: ventana de palabras, análisis de dependencias sintácticas y un enfoque híbrido que combina ambas técnicas. Se emplean los mismos diez modelos utilizados para la tarea de extracción de aspectos, basados en arquitecturas *Transformers* para idioma español. A continuación, en la , se detalla los hiperparámetros comunes utilizados en todos los experimentos.

Tabla 20. Configuración de los hiperparámetros de los modelos

|  |  |
| --- | --- |
| **Hiperparámetros** | **Valor** |
| Semilla (*seed*) | 42 |
| Tasa de aprendizaje | 2e-5 |
| Cantidad de épocas (*epochs*) | 5 |
| Tamaño de lote | 8 |
| Tamaño máximo de la secuencia de palabras | 512 |

**Método de ventana de palabras**

Se exploran ventanas de tamaño variable entre 1 y 7 palabras alrededor del aspecto, para determinar el mejor contexto. A continuación, en la , se observa que el tamaño de ventana 6 muestra el mejor equilibrio entre precisión (94.07%) y recall (97.33%) en BERTIN-base, alcanzando un F1-score de 95.67%. Este método aprovecha la proximidad léxica para capturar términos de opinión relevantes, como adjetivos o adverbios asociados al aspecto.

Tabla 21. Análisis de los resultados de diferentes tamaño de ventana.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bertin\_Base | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Accuracy | 87.98 | 90.91 | 92.18 | 93.06 | 93.35 | **93.55** | **93.35** |
| Precision | 88.18 | 91.21 | 92.40 | 93.24 | 93.71 | **94.07** | 94.05 |
| Recall | 96.53 | 96.93 | 97.33 | **97.60** | 97.46 | 97.33 | 97.06 |
| F1 | 92.17 | 93.98 | 94.80 | 95.37 | 95.55 | **95.67** | 95.53 |

Por otra parte, en la , se destacan los modelos BERTIN-large y BETO en la métrica F1. Sin embargo, GPT-2 obtuvo el peor rendimiento, debido a su diseño autoregresivo.

Tabla 22. Resultados de polaridad con el método de ventana de palabras (tamaño 6)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelos | Accuracy | Precisión | Recall | F1 |
| Beto | 91.50 | 92.33 | 96.40 | 94.32 |
| Bert | 89.45 | 91.25 | 94.66 | 92.93 |
| Albert\_base | 90.33 | 91.15 | 96.13 | 93.57 |
| **Bertin\_base** | **93.55** | **94.07** | **97.33** | **95.67** |
| Bertin\_large | 93.16 | 93.70 | 97.20 | 95.41 |
| Electra\_small | 80.76 | 82.41 | 93.73 | 87.71 |
| Electra\_base | 86.23 | 88.01 | 94.00 | 90.90 |
| Gpt\_2 | 79.19 | 81.77 | 92.13 | 86.64 |

**Método de dependencias sintácticas**

Este enfoque identifica relaciones gramaticales entre el aspecto y los términos de opinión. Aunque se considera útil para capturar conexiones estructurales, los resultados fueron inferiores al método anterior. A continuación, en la , se observa que BERTIN-large logra el mejor F1-score, seguido de BERTIN-base, mientras que GPT-2 nuevamente muestra limitaciones. La menor precisión general sugiere que las dependencias sintácticas, pueden omitir términos de opinión contextuales no directamente vinculados.

Tabla 23. Resultados de polaridad con el método de análisis de dependencias

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelos | Accuracy | Precisión | Recall | F1 |
| Beto | 88.88 | 89.84 | 95.72 | 92.69 |
| Bert | 86.52 | 88.34 | 94.12 | 91.14 |
| Albert\_base | 87.80 | 89.70 | 94.25 | 91.92 |
| Bertin\_base | 90.36 | 91.35 | **95.99** | 93.61 |
| Bertin\_large | **90.95** | **92.60** | 95.32 | **93.94** |
| Electra\_small | 79.94 | 82.87 | 91.72 | 87.07 |
| Electra\_base | 82.59 | 85.66 | 91.72 | 88.58 |
| Gpt\_2 | 76.10 | 81.46 | 87.44 | 84.35 |

**Enfoque híbrido**

La combinación de ventana de palabras (tamaño 6) y dependencias sintácticas permite aprovechar las ventajas de ambos métodos. Como se aprecia en la , BERTIN-base obtiene el mejor rendimiento, seguido de BERTIN-large. Este enfoque mitiga falsos negativos al integrar contexto local y relaciones sintácticas, mejorando el recall sin sacrificar precisión.

Tabla 24. Resultados de polaridad con el método hibrido

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelos | Accuracy | Precisión | Recall | F1 |
| Beto | 91.01 | 91.96 | 96.13 | 94.00 |
| Bert | 88.18 | 90.79 | 93.33 | 92.04 |
| Albert\_base | 89.64 | 91.17 | 95.06 | 93.08 |
| Bertin\_base | **92.96** | 93.57 | **97.06** | **95.28** |
| Bertin\_large | 92.87 | **94.36** | 96.00 | 95.17 |
| Electra\_small | 80.85 | 82.81 | 93.20 | 87.70 |
| Electra\_base | 86.13 | 88.19 | 93.60 | 90.81 |
| Gpt\_2 | 78.71 | 83.00 | 89.20 | 85.98 |

### Análisis de los resultados de la clasificación de la polaridad.

Tras evaluar los diferentes mecanismos de selección de contexto y modelos de lenguaje para la clasificación de la polaridad, se determina que la combinación de Bertin\_base y una ventana de palabras de tamaño 6 ofrecen el mejor rendimiento. Como se muestra en la , esta solución supera los resultados reportados en la literatura, con mayor calidad que los de la competición SemEval 2016 Task 5 [28].

Tabla 25. Comparación de la solución propuesta para clasificación de polaridad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Soluciones** | **Accuracy** | **Precisión** | **F1** | **Recall** |
| IIT-T/U\* | **83.50** |  |  |  |
| TGB/C\* | **82.00** |  |  |  |
| UWB/C\* | 81.30 |  |  |  |
| INSIG/C\* | **79.50** |  |  |  |
| SemEval-2016 – baseline\* | 77.80 |  |  |  |
| [[6](#_ENREF_39)] | 84.70 |  |  |  |
| [[22](#_ENREF_22)] | 0.866 | 0.836 | 0.864 | 0.848 |
| [[39](#_ENREF_39)] | 84.80 |  |  |  |
| [[40](#_ENREF_40)] | 79.69 |  |  |  |
| [[58](#_ENREF_40)] | 87.10 |  |  |  |
| **Bertin\_base + Ventana de palabras (tamaño 6)** | **93.55** | **94.07** | **97.33** | **95.67** |

Estos resultados destacan la eficacia de los modelos de lenguaje basados en *Transformers* para el análisis de sentimiento basado en aspectos, y la importancia de una adecuada selección del contexto del aspecto para optimizar la precisión.

A diferencia de las soluciones comparadas, se incluye en esta evaluación las métricas de accuracy, precisión, recall y F1-score, proporcionando una descripción más completa del desempeño del modelo.

## Conclusiones parciales.

Tras la evaluación exhaustiva de la solución propuesta para el análisis de sentimientos basado en aspectos en español, se pueden arribar a las siguientes conclusiones:

* La combinación de modelos mediante votación máxima (BETO, BERT multilingüe, ALBERT-base, ALBERT-large y ALBERT-xx-large) demuestra ser superior a los enfoques individuales, alcanzando un F1-score del 86.45% en la extracción de aspectos. Esto valida que el ensamblaje mitiga sesgos y aprovecha las fortalezas de cada arquitectura, superando significativamente a soluciones previas reportadas para el mismo conjunto de datos(SemEval 2016).
* Los modelos como BETO y BERTIN-base muestran un rendimiento destacado, especialmente en la clasificación de polaridad. Esto resalta la importancia de utilizar modelos pre-entrenados específicamente en español para tareas de PLN, ya que capturan mejor las particularidades lingüísticas frente a alternativas multilingües o genéricas.
* La estrategia de ventana de palabras (tamaño 6) fue la más efectiva para identificar términos de opinión, logrando un mejor equilibrio. Esto resalta la importancia del contexto local alrededor de los aspectos, aunque el enfoque híbrido (ventana + dependencias sintácticas) puede ser apropiado para casos con estructuras gramaticales complejas.
* GPT-2, diseñado para generación de texto, obtiene los peores resultados, por lo que se considera inadecuado para tareas de etiquetado secuencial.

# Conclusiones generales.

Como resultado de esta investigación se desarrolló un método para la extracción de aspectos con el uso de modelos Transformers se aplica un enfoque de ensamble de los clasificadores para extraer los aspectos de las opiniones de entrada. Cumpliéndose de esta forma el objetivo general propuesto, ya que:

* La revisión de la literatura revela una escasez de soluciones para el análisis de sentimientos basado en aspectos en el idioma español. Una de las causas principales de esta limitación es la falta de conjuntos de datos etiquetados adecuados para entrenar modelos en este idioma.
* La revisión exhaustiva de la literatura y el análisis de las fortalezas y limitaciones de las soluciones actuales indican que la aplicación del **aprendizaje por transferencia** ofrece resultados prometedores en lenguajes con recursos lingüísticos limitados para el análisis de sentimientos basado en aspectos.
* La utilización de los modelos de lenguaje con arquitectura Transformers permite superar el estado del arte para la subtarea de extracción de aspectos para idioma español.
* La utilización de técnicas de ensamble demuestran que, para esta tarea de extracción de aspectos con el conjunto de datos de SemEval 2016 se obtienen mejores resultados que los modelos base independientes.

# Recomendaciones.

* La recopilación de un corpus más amplio de reseñas en español, abarcado distinto dominios, permitirá explorar la aplicabilidad de la solución más allá del dominio de restaurantes donde se ha validado.
* Se sugiere explorar alternativas para la representación de los aspectos, con el objetivo de mejorar el entrenamiento de los modelos de lenguaje.
* Se recomienda investigar el impacto del entrenamiento con datos etiquetados en inglés y español, en un modelo multilingüe, para evaluar si esto puede mejorar la calidad de los resultados.
* Se recomienda extender la evaluación de la solución propuesta a un contexto real, más allá del entorno controlado de la investigación.
* Las pruebas con otros tipos de técnicas de ensamble son fundamental para lograr un estudio más profundo y evaluar sus resultados para compararlos con la técnica utilizada de votación máxima.
* Ampliar los modelos base de Transformers en busca de nuevas arquitecturas

# Referencias Bibliográficas

[1] Y. Wu, Z. Jin, C. Shi, P. Liang, and T. Zhan, "Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis," *arXiv preprint arXiv:2403.08217,* 2024.

[2] B. Liu, "Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers," 2012.

[3] B. Liu, *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[4] Y. Montaño Reyes, "Análisis de sentimiento de mensajes de correo electrónico bajo un enfoque de transfer learning," 2022.

[5] M. Shams, N. Khoshavi, and A. Baraani-Dastjerdi, "LISA: language-independent method for aspect-based sentiment analysis," *IEEE Access,* vol. 8, pp. 31034-31044, 2020.

[6] C. H. Miranda and E. Buelvas, "AspectSA: Unsupervised system for aspect based sentiment analysis in Spanish," *Prospectiva,* vol. 17, no. 1, pp. 87-95, 2019.

[7] A. P. Pandian, "Performance evaluation and comparison using deep learning techniques in sentiment analysis," *Journal of Soft Computing Paradigm (JSCP),* vol. 3, no. 02, pp. 123-134, 2021.

[8] D. López Ramos and L. Arco García, "Aprendizaje profundo para la extracción de aspectos en opiniones textuales," *Revista Cubana de Ciencias Informáticas,* vol. 13, no. 2, pp. 105-145, 2019.

[9] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimedia tools and applications,* vol. 82, no. 3, pp. 3713-3744, 2023.

[10] J. Jia, W. Liang, and Y. Liang, "A review of hybrid and ensemble in deep learning for natural language processing," *arXiv preprint arXiv:2312.05589,* 2023.

[11] S. Arroni, Y. Galán, X. M. Guzmán Guzmán, E. R. Núñez Valdéz, and A. Gómez Gómez, "Sentiment analysis and classification of hotel opinions in twitter with the transformer architecture," *International Journal of Interactive Multimedia and Artificial Intelligence,* 2023.

[12] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems,* 2017.

[13] A. Mohammed and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," *Journal of King Saud University-Computer and Information Sciences,* vol. 35, no. 2, pp. 757-774, 2023.

[14] I. D. Mienye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access,* vol. 10, pp. 99129-99149, 2022.

[15] K. Chowdhary and K. Chowdhary, "Natural language processing," *Fundamentals of artificial intelligence,* pp. 603-649, 2020.

[16] E. D. Liddy, "Natural language processing," 2001.

[17] G. J. R. Fernández, "Método no supervisado para la detección de polaridad en opiniones," *Trabajo de Diploma, Universidad Tecnológica de La Habana “José Antonio Echeverría,* vol. La Habana, Cuba, 2022.

[18] R. O. A. Ayala, "Método para la detección de polaridad en opiniones: un enfoque híbrido y aplicando ensamble de clasificadores,," *Trabajo de diploma presentado en opción del título de Ingeniero Informático, Universidad Tecnológica de La Habana “José Antonio Echeverría,* 2022.

[19] Y. S. Mehanna and M. B. Mahmuddin, "A semantic conceptualization using tagged bag-of-concepts for sentiment analysis," *IEEE access,* vol. 9, pp. 118736-118756, 2021.

[20] D. López and L. Arco, "Multi-domain aspect extraction based on deep and lifelong learning," in *Iberoamerican Congress on Pattern Recognition*, 2019: Springer, pp. 556-565.

[21] A. Karimi, L. Rossi, and A. Prati, "Adversarial training for aspect-based sentiment analysis with bert," in *2020 25th international conference on pattern recognition (ICPR)*, 2021: IEEE, pp. 8797-8803.

[22] P. Montañez Castelo, " Sistema de análisis de sentimientos basado en aspectos para idioma español," 2023.

[23] M. M. Truşcǎ and F. Frasincar, "Survey on aspect detection for aspect-based sentiment analysis," *Artificial Intelligence Review,* vol. 56, no. 5, pp. 3797-3846, 2023.

[24] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in *Third IEEE international conference on data mining*, 2003: IEEE, pp. 427-434.

[25] E. M. Aboelela, W. Gad, and R. Ismail, "Ontology-Based Approach for Feature Level Sentiment Analysis," *International Journal of Intelligent Computing and Information Sciences,* vol. 21, no. 3, pp. 1-12, 2021.

[26] A. García-Pablos, M. Cuadros, and G. Rigau, "W2VLDA: almost unsupervised system for aspect based sentiment analysis," *Expert Systems with Applications,* vol. 91, pp. 127-137, 2018.

[27] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology (IJCTT),* vol. 48, no. 3, pp. 128-138, 2017.

[28] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *International workshop on semantic evaluation*, 2016, pp. 19-30.

[29] L. Fröhling and A. Zubiaga, "Feature-based detection of automated language models: tackling GPT-2, GPT-3 and Grover," *PeerJ Computer Science,* vol. 7, p. e443, 2021.

[30] Y. Mao, Y. Shen, C. Yu, and L. Cai, "A joint training dual-mrc framework for aspect based sentiment analysis," in *Proceedings of the AAAI conference on artificial intelligence*, 2021, vol. 35, no. 15, pp. 13543-13551.

[31] D. Mullick, "Studying Limitations of Generative Transformer based models for Aspect Based Sentiment Analysis," 2023.

[32] Ł. Augustyniak, T. Kajdanowicz, and P. Kazienko, "Comprehensive analysis of aspect term extraction methods using various text embeddings," *Computer Speech & Language,* vol. 69, p. 101217, 2021.

[33] L. De Mattei, G. De Martino, A. Iovine, A. Miaschi, M. Polignano, and G. Rambelli, "ATE ABSITA@ EVALITA2020: Overview of the aspect term extraction and aspect-based sentiment analysis task," in *CEUR WORKSHOP PROCEEDINGS*, 2020, vol. 2765: CEUR-WS, pp. 67-74.

[34] A. Gupta and U. Sharma, "Deep Learning-Based Aspect Term Extraction for Sentiment Analysis in Hindi," *Indian Journal of Science and Technology,* vol. 17, no. 7, pp. 625-634, 2024.

[35] H. Xu, B. Liu, L. Shu, and P. S. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," *arXiv preprint arXiv:1904.02232,* 2019.

[36] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Dombert: Domain-oriented language model for aspect-based sentiment analysis," *arXiv preprint arXiv:2004.13816,* 2020.

[37] H. Yan, J. Dai, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," *arXiv preprint arXiv:2106.04300,* 2021.

[38] M. Patel and C. I. Ezeife, "BERT-based multi-task learning for aspect-based opinion mining," in *International conference on database and expert systems applications*, 2021: Springer, pp. 192-204.

[39] C. Henríquez, F. Briceño, and D. Salcedo, "Unsupervised model for aspect-based sentiment analysis in Spanish," *IAENG International Journal of Computer Science,* vol. 46, no. 3, pp. 430-438, 2019.

[40] B.-C. Martínez-Seis, O. Pichardo-Lagunas, S. Miranda, I.-J. Perez-Cazares, and J.-A. Rodriguez-González, "Deep learning approach for aspect-based sentiment analysis of restaurants reviews in Spanish," *Computación y Sistemas,* vol. 26, no. 2, pp. 899-908, 2022.

[41] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[42] I. C. González, "con BERT: Análisis de sentimientos en tuits”."

[43] J. F. Auquilla Vicuña and J. C. Mora Alvarez, "Diseño de un sistema prototipo de diálogo persona-máquina basado en la arquitectura BERT," 2022.

[44] H. Wang, X. Hu, and H. Zhang, "Sentiment analysis of commodity reviews based on ALBERT-LSTM," in *Journal of Physics: Conference Series*, 2020, vol. 1651, no. 1: IOP Publishing, p. 012022.

[45] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "RoBERTa-LSTM: a hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access,* vol. 10, pp. 21517-21525, 2022.

[46] G. R. Narayanaswamy, "Exploiting BERT and RoBERTa to improve performance for aspect based sentiment analysis," 2021.

[47] E. T. R. Schneider, J. V. A. de Souza, Y. B. Gumiel, C. Moro, and E. C. Paraiso, "A GPT-2 language model for biomedical texts in Portuguese," in *2021 IEEE 34th international symposium on computer-based medical systems (CBMS)*, 2021: IEEE, pp. 474-479.

[48] S. Alrowili and K. Vijay-Shanker, "ArabicTransformer: Efficient large Arabic language model with funnel transformer and ELECTRA objective," in *Findings of the association for computational linguistics: EMPLN 2021*, 2021, pp. 1255-1261.

[49] E.-S. Apostol, A.-G. Pisică, and C.-O. Truică, "ATESA-B {\AE} RT: A Heterogeneous Ensemble Learning Model for Aspect-Based Sentiment Analysis," *arXiv preprint arXiv:2307.15920,* 2023.

[50] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward, "Improving predictions using ensemble Bayesian model averaging," *Political Analysis,* vol. 20, no. 3, pp. 271-291, 2012.

[51] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern recognition,* vol. 36, no. 12, pp. 2757-2767, 2003.

[52] G.-R. Latif-Shabgahi, "A novel algorithm for weighted average voting used in fault tolerant computing systems," *Microprocessors and Microsystems,* vol. 28, no. 7, pp. 357-361, 2004.

[53] C. Soares, P. B. Brazdil, and P. Kuba, "A meta-learning method to select the kernel width in support vector regression," *Machine learning,* vol. 54, pp. 195-209, 2004.

[54] L. Breiman, "Bagging predictors," *Machine learning,* vol. 24, pp. 123-140, 1996.

[55] J. V. Román, E. M. Cámara, J. G. Morera, and S. M. J. Zafra, "Tass 2014-the challenge of aspect-based sentiment analysis," *Procesamiento del Lenguaje Natural,* vol. 54, pp. 61-68, 2015.

[56] G. H. Zarate Calderon, "Análisis de sentimiento en información de medios periodísticos y redes sociales mediante redes neuronales recurrentes," 2021.

[57] T. Granollers, J. Navarro, J. Suarez, and M. Rodríguez, "Perspectivas en la interacción humano-tecnología," *Jaime Muñoz-Arteaga, César A. Collazos, Toni Granollers, Huizilopoztli Luna-García,* vol. 39, 2022.