DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA CLASIFICACIÓN DE RESEÑAS EN PORTALES WEB UTILIZANDO ANÁLISIS DE SENTIMIENTOS Y MODELOS OCULTOS DE MARKOV

Leidy Tatiana Llanos Gallego
Código 8986550
John Díaz Alonso
Código 8961548

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director
Cristhian Kaori Valencia Marín, MEng

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, ENERO 19 DE 2026

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

INTRODUCCIÓN

Las reseñas se han convertido en un insumo relevante y beneficioso para las empresas y comercios, debido al crecimiento de los *ecommerce*, ya que generan un posicionamiento de una marca, artículo o imagen de una compañía, aumentando las ventas y la confianza de los clientes, puesto que una reseña se describe como un escrito breve que informa y a la vez valora un producto o servicio, cabe destacar que una reseña tiene una característica fundamental, y es que radica en describir y emitir un juicio valorativo a favor o en contra [1], lo cual genera la necesidad de realizar un análisis automatizado, categorizar las reseñas y las valoraciones de usuarios de los portales web, y así buscar e identificar patrones en los datos y comprender las opiniones expresadas.

Por otro lado, en estadísticas recientes, el 95% de los clientes leen reseñas en línea antes de realizar una compra y el 93% de los consumidores afirman que las reseñas en línea influyen en sus decisiones de compra, por lo cual es evidente que la opinión de otras personas es un factor clave en el proceso de ventas [2].

En este contexto, se vuelve esencial contar con técnicas que permitan clasificar, extraer y detectar actitudes, sentimientos y opiniones expresadas en forma textual sobre diversos productos o servicios, con la finalidad de favorecer todo tipo de negocios, industrias, o emprendimientos, que pretendan optimizar la reputación de su marca, la satisfacción del consumidor, con estrategias de inteligencia de mercado, imponiéndose y aventajando a sus competidores [3].

En consecuencia, y a partir del escenario descrito, se estableció el requerimiento de desarrollar un sistema inteligente que identifique y clasifique las opiniones y emociones de las reseñas de usuarios mediante modelos ocultos de Markov, proporcionando una retroalimentación directa para la toma de decisiones informadas y así mejorar eficazmente la toma de decisiones de la gerencia, para lo cual se incluyó la implementación de una metodología para el preprocesamiento de datos, el desarrollo y ejecución de un modelo oculto de Markov, y asimismo pruebas exhaustivas del sistema con conjuntos de datos reales. Adicionalmente, se da la relevancia estratégica de clasificar opiniones del comercio electrónico y se resalta que el proyecto aportó un desarrollo en minería de texto, procesamiento del lenguaje natural y modelos estadísticos como los procesos de Márkov con parámetros desconocidos.

Por esta razón, se llevó a cabo el proyecto que desarrolló un sistema inteligente que identifique y clasifique opiniones y emociones en reseñas de usuarios en portales web, por lo tanto, se propuso una metodología de preprocesamiento de datos y la implementación de un Modelo Oculto de Márkov para capturar estructuras latentes y transiciones de estados de ánimo en las reseñas. Lo anterior, a través de la construcción de un módulo de software en Python para la clasificación de reseñas en portales web a partir de modelos ocultos de markov y procesamiento de lenguaje natural. Las pruebas se realizaron con conjuntos de datos reales, evaluando la precisión y eficacia del modelo en términos de clasificación, utilizando una base de datos etiquetada y técnicas de análisis de texto, junto con algoritmos de Baum-Welch para la estimación de parámetros y la implementación de

Modelos Ocultos de Márkov.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En el año 1991 se publicó el primer sitio web, desarrollado por Tim Berners-Lee en el CERN (Organización Europea para la Investigación Nuclear), lo cual marcó el nacimiento de la World Wide Web. A partir de ese momento, su crecimiento ha sido exponencial, lo que permitió la aparición de diversos tipos de sitios como los de comercio electrónico, educativos, blogs, portales e institucionales, entre otros [4].

Además, la evolución de Internet, analizada en enero de 2023, se traduce en un total de 1.132.268.801 páginas web, 270.967.923 dominios únicos y 12.156.700 servidores, las cuales, en la mayoría de los casos, son para establecer tiendas online, desarrollar estrategias de marketing digital o establecer comunicación con los usuarios [5].

Es más, las reseñas en línea se han consolidado como una herramienta clave en los sitios web, ya que permiten a los usuarios expresar sus opiniones, valoraciones y experiencias sobre productos o servicios. Esta interacción, conocida como "sentimiento del usuario", constituye un insumo valioso para las organizaciones al momento de identificar percepciones, necesidades y actitudes de los consumidores [6].

Por consiguiente, los comentarios de los usuarios se reflejan como un insumo importante descritos por medio de reseñas, las cuales están definida como colecciones de cadenas de texto, y descritas en oraciones, que deben ser extraídos automáticamente del portal web, con su valoración de estrellas o calificaciones, identificando el sentimiento de la reseña por parte del usuario [7], adicionalmente, en el marco del auge de las herramientas basadas en inteligencia artificial, es necesario y pertinente el desarrollo de sistemas inteligentes y métodos automatizados que permitan generar y optimizar una comprensión categorizada sobre los sentimientos y la postura de los usuarios, así como analizar, reconocer y comprender sentimientos y emociones en el texto; lo anterior se traduce en una ventaja competitiva para estos portales, permitiendo a sus creadores una retroalimentación directa que beneficia la toma de decisiones informadas y a la extracción de conocimiento sobre opiniones subjetivas de los usuarios, logrando así obtener un beneficio al disponer de una herramienta automatizada [8].

En el aprendizaje automático (*machine learning*), una de las ramas de la inteligencia artificial, existen algoritmos que presentan un rendimiento deficiente en el análisis de sentimientos. En la revista de Ingeniería UC se publicó un artículo con el título, "Un sistema híbrido basado en modelos ocultos de Markov y máquinas de vectores de soporte con aprendizaje hacia adelante para reconocimiento de fonos en habla continua venezolana"[9], en el que se evaluó el rendimiento de un reconocedor automático del habla basado en modelos ocultos de Markov y máquinas de vectores de soporte, se demostró que las máquinas de vectores de soporte, experimentan limitaciones debido a que trabajan con vectores de entrada de dimensión fija [9], de igual modo, en la tesis denominada "Estudio de los modelos ocultos de Markov y desarrollo de un prototipo para

el reconocimiento automático del habla" de la Universidad Politécnica Salesiana, el cual tiene como objetivo principal estudiar los modelos ocultos de markov y desarrollo de un prototipo para el reconocimiento automático del habla, en donde, realizaron el entrenamiento mediante redes neuronales artificiales y modelos ocultos de Markov, y en el caso de los algoritmos de redes neuronales, necesitan de tiempo y dedicación en el momento de realizar los entrenamientos, adicionalmente solo tienen la capacidad de procesamiento espacial [10], lo que dificulta el análisis y se es necesario, modelos deterministas que describen las series de tiempo e identifiquen los instantes de tiempo y detallen las series empleando algunas propiedades específicas [11], con la finalidad de analizar concretamente las reseñas, las opiniones y emociones expresadas por el usuario.

1.2. FORMULACIÓN DEL PROBLEMA

a. ¿Cómo realizar un análisis automatizado y categorizado, empleando métodos estadísticos y análisis de datos sobre las reseñas y las valoraciones que hacen los usuarios en los portales web, para la identificación de patrones en los datos y la comprensión de las opiniones?

   - ¿Cómo procesar los textos para interpretar, analizar y obtener mejores resultados para los métodos estadísticos y de análisis de los datos?
   - ¿Cómo construir un modelo oculto de Markov que permita realizar un análisis de sentimientos clasificando las reseñas de los usuarios?
   - ¿Cómo evaluar el nivel de rendimiento de los métodos empleados?

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Desarrollar un sistema inteligente que identifique y clasifique las opiniones y emociones expresadas en las reseñas de los usuarios de los portales web, por medio modelos ocultos de Markov, para obtener una ventaja competitiva y una retroalimentación directa que beneficie la toma de decisiones informadas.

2.2. OBJETIVOS ESPECÍFICOS

- Implementar una metodología para el preprocesamiento de los datos provenientes de los portales web, realizando el tratamiento a las cadenas textos o reseñas, por medio de modelos de procesamiento del lenguaje natural.
- Desarrollar e implementar un modelo oculto de Márkov que capture las estructuras latentes y las transiciones de estados de ánimo en las reseñas de portales web, permitiendo la identificación precisa de los sentimientos expresados por los usuarios.
- Realizar pruebas y evaluaciones exhaustivas del sistema desarrollado utilizando un conjunto de datos reales de portales web, evaluando la precisión y eficacia en términos de clasificación.

2.3. RESULTADOS ESPERADOS

- Un módulo de software en Python para el procesamiento de base de datos de portales web.
- Un módulo de software en Python para la clasificación de reseñas en portales web a partir de modelos ocultos de Markov y procesamiento de lenguaje natural.
- Documento con el desarrollo del proyecto y análisis de los resultados.

3. MARCO TEÓRICO Y ANTECEDENTES

3.1. MARCO TEÓRICO

3.1.1. METODOLOGÍA GENERAL DEL PROYECTO

En la siguiente tabla se expone el resumen detallado de las actividades metodológicas desarrolladas durante la ejecución del proyecto, en el que se estructura y condensan las fases, tareas y resultados obtenidos en el diseño e implementación del sistema de clasificación de reseñas mediante técnicas de análisis de sentimientos y Modelos Ocultos de Markov.

En el cual, se resumen las etapas metodológicas con el proceso investigativo, desde la recolección, preparación y procesamiento del lenguaje natural (PLN) hasta la modelación probabilística, entrenamiento, validación y evaluación de la precisión del sistema.

FASES	ACTIVIDADES METOLOGÍCAS
PROCESAMIENTO DE DATOS Y TRATAMIENTO DE CADENAS DE TEXTOS Base textual limpia, normalizada y estructurada para modelamiento probabilístico.	1. Recolección de la Base de Datos Recopilación de reseñas del portal web Sentiment Polarity Annotations Dataset (SPOT), que incluye opiniones positivas, neutras y negativas. Selección de datasets Yelp e IMDb, por su volumen y diversidad en contenido textual.
	2. Exploración de los Set de Datos Análisis de la distribución de las reseñas en cada dataset que identifica el equilibrio de clases y la naturaleza de los textos, con tendencias en la cantidad de reseñas por tipo de sentimiento y generación de nubes de palabras y gráficos de distribución para explorar términos frecuentes y patrones lingüísticos.
	3. Selección de los Datos Etiquetados Filtración de las reseñas, conservando solo las etiquetas (+) positivas, (Ø) neutrales y (-) negativas, con eliminación de otras clasificaciones numéricas y datos irrelevantes mediante funciones de limpieza en Pandas, garantizando consistencia en la representación del sentimiento.
	4. Preparación de los Datos Ejecución de procesos: – Eliminación de caracteres especiales y duplicados. – Corrección de errores ortográficos. – Conversión de texto a minúsculas. – Eliminación de signos de puntuación. – Verificación de datos faltantes (NaN), confirmando integridad total de los datasets.
	5. Procesamiento del Lenguaje Natural (PNL)

	Aplicación de técnicas de NLP: - Tokenización: división de las reseñas en unidades léxicas. - Eliminación de stopwords: exclusión de palabras sin valor semántico. - Lematización: reducción de palabras a su forma base. - Etiquetado POS: clasificación de palabras por categoría gramatical para enriquecer la representación lingüística.
MODELADO A PARTIR DE MODELOS PROBABILÍSTICOS DE DATOS ASOCIADOS CON RESEÑAS Modelo capaz de clasificar reseñas según sentimiento con métricas de rendimiento (Accuracy, F1-Score).	1. Técnicas de Extracción de Características Utilización de métodos de vectorización del texto como Bag of Words (BoW), TF-IDF, y Word Embeddings, GloVe 300d, Word2Vec-384d, FastText-300 y BERT-base para convertir el lenguaje natural en representaciones numéricas que puedan ser procesadas por los Modelos Ocultos de Markov.
	2. Estimación de Parámetros para el Modelo Selección de parámetros del HMM como las probabilidades de transición y emisión por medio de experimentos, asegurando una correcta representación de las secuencias textuales con evaluación de diferentes covarianzas (diag, full, tied, spherical) bajo criterios AIC y BIC.
	3. Selección del Número de Estados Ocultos (AIC/BIC) Ejecución de simulaciones con distintos valores de n_components (número de estados ocultos). - Las métricas AIC y BIC se usaron para identificar los modelos con menor penalización y mejor ajuste. - Determinación de rangos óptimos de estados ocultos donde ambos criterios convergieron o alcanzaron mínimos locales estables, indicando la mejor complejidad del modelo.
	4. Entrenamiento del Modelo Propuesto Acciones: - Mapeo de sentimientos: vinculación entre etiquetas textuales y estados del modelo. - División de datos: separación entre entrenamiento y validación. - Definición del HMM: estructura de estados, emisiones y transiciones probabilísticas.
	5. Implementación del Modelo en Python Implementación utilizando Python, integrando módulos de NLP y HMM. - Entrenamiento mediante el algoritmo de Baum-Welch (estimación de parámetros). - Predicción de secuencias ocultas con el algoritmo de Viterbi.

EVALUACIÓN DE LA PRECISIÓN Y EFICACIA DEL SISTEMA DESARROLLADO
 	
Modelo con eficacia y precisión, y viabilidad del uso de Modelos Ocultos de Markov en el análisis de sentimientos de reseñas en portales web.
	1. Evaluación de Modelo
	
	Evaluación del desempeño del modelo implementado con diferentes configuraciones y datasets.
	
	Actividades principales
	
	- Comparación de resultados de clasificación entre datasets IMDb y Yelp.
	- Cálculo de métricas de rendimiento:
	  o Accuracy: mide la proporción de clasificaciones correctas.
	  o F1-Score: balance entre precisión y exhaustividad.

	2. Análisis de Resultados - Los mejores modelos alcanzaron valores de F1-Score aproximados a 0.75, y Accuracy superior a 0.85, proporcionan un desempeño competitivo en tareas de análisis de sentimientos. - El desempeño varió según la técnica de vectorización y la cantidad de estados ocultos. - Evidencia de equilibrio entre complejidad del modelo y capacidad de generalización.

Tabla 1: Resumen detallado las actividades metodológicas

3.1.2. PORTALES WEB Y RESEÑAS EN INTERNET

Un portal Web es un sitio de Internet caracterizado por facilitar el acceso a distintos recursos o servicios de la world wide web. Estos accesos pueden manejar temas relacionados o ser de diversa índole, ofreciendo así un amplio abanico de temas que podrían ser de interés para el usuario [12].

Una reseña es una valoración, opinión o comentario sobre un servicio, producto o experiencia concreta, así como sobre cualquier negocio en general. Las reseñas no dejan de ser como las recomendaciones de amigos o familiares, pero publicadas en la red por completos desconocidos. Aunque la simple valoración de un usuario en el enorme mundo de Internet parezca algo banal, la realidad es que las reseñas tienen grandes consecuencias para las empresas [13], como:

- En primer lugar, y la más importante, es que ejercen una fuerte influencia en el comportamiento y el proceso de decisión de la población conectada.
- En la mayoría de las ocasiones, las reseñas también influyen en la imagen de marca de la empresa, así como en su reputación.
- Son fuente de información directa para las empresas, dando las oportunidades necesarias para la mejora del negocio.
- En último lugar, y no por ello menos importante, en ciertas ocasiones las reseñas pueden mejorar la visibilidad de tu empresa en Internet.

3.1.3. PROCESAMIENTO DEL LENGUAJE NATURAL - PNL

Inicialmente, el lenguaje natural se refiere a un lenguaje que los humanos utilizan para la comunicación cotidiana; Idiomas como inglés, hindi o portugués. A diferencia de los lenguajes artificiales, como los lenguajes de programación y las notaciones matemáticas, los lenguajes naturales han evolucionado a medida que pasan de generación en generación y son difíciles de definir con reglas explícitas [14].

El Procesamiento del Lenguaje Natural (o PNL *Natural Language Processing*) utiliza el aprendizaje automático para procesar e interpretar textos y datos. En un extremo, podría ser tan simple como contar la frecuencia de las palabras para comparar diferentes estilos de escritura [15].

En el otro extremo, la PNL implica "comprender" expresiones humanas completas, hasta el punto de poder darles respuestas útiles. Las tecnologías basadas en PNL están cada vez más extendidas. Por ejemplo, los teléfonos y las computadoras portátiles admiten texto predictivo y reconocimiento de escritura a mano; los motores de búsqueda web dan acceso a información contenida en texto no estructurado; La traducción automática nos permite recuperar textos escritos en chino y leerlos en español. Al proporcionar interfaces hombre-máquina más naturales y un acceso más sofisticado a la información almacenada, el procesamiento del lenguaje ha pasado a desempeñar un papel central en la sociedad de la información multilingüe [15].

3.1.4. MINERÍA DE TEXTO Y ANÁLISIS DE SENTIMIENTO

La minería de texto, también conocido como análisis de texto, es el proceso de extracción de patrones significativos y conocimientos de grandes conjuntos de datos de texto utilizando técnicas de procesamiento del lenguaje natural (PNL), estadísticas y aprendizaje automático. El análisis de sentimiento o minería de opinión es una subdisciplina de la minería de texto que se centra en identificar y clasificar las opiniones y emociones expresadas en los datos de texto. Este texto explora la minería de texto y el análisis de sentimiento, abordando conceptos clave, técnicas y aplicaciones en diversos campos [16].

La minería de texto implica el uso de algoritmos y técnicas para analizar y procesar datos de texto con el fin de descubrir patrones, tendencias y relaciones ocultas en los datos. El proceso de minería de texto generalmente involucra tres etapas principales: preprocesamiento de texto, representación de texto y análisis de texto. El preprocesamiento de texto incluye la limpieza, normalización y tokenización de los datos de texto. La representación de texto implica convertir los datos de texto en formatos numéricos que puedan ser procesados por algoritmos de aprendizaje automático y estadísticos, como la bolsa de palabras, la frecuencia de término documento (TF-IDF) y la incrustación de palabras [17].

El análisis de texto se refiere a la aplicación de técnicas y algoritmos para extraer información útil y conocimientos de los datos de texto procesados y representados. El análisis de sentimiento puede ser clasificado en enfoques basados en léxico, enfoques de aprendizaje supervisado y enfoques de aprendizaje no supervisado. Los enfoques basados en léxico utilizan listas de palabras predefinidas o diccionarios de sentimientos para

determinar el sentimiento de un texto. Los enfoques de aprendizaje supervisado emplean algoritmos de aprendizaje automático para clasificar los sentimientos en función de un conjunto de datos etiquetado previamente. Los enfoques de aprendizaje no supervisado utilizan técnicas como el agrupamiento para identificar patrones y estructuras en los datos de sentimiento sin necesidad de datos etiquetados [17].

3.1.5. CRITERIO DE INFORMACIÓN DE AKAIKE (AIC) Y CRITERIO DE INFORMACIÓN BAYESIANO (BIC)

3.1.5.1. Criterio de Información de Akaike (AIC)

El Criterio de Información de Akaike (AIC) es una medida que se utiliza para evaluar modelos equilibrando el ajuste y la complejidad. Recibe su nombre del estadístico japonés Hirotugu Akaike, quien lo desarrolló para ayudar a seleccionar el mejor modelo sin sobreajuste [34], por lo que el criterio:

- Considera la capacidad del modelo para ajustarse a los datos.
- Penaliza el modelo por cada parámetro adicional para desalentar el sobreajuste.

La cual se describe en la siguiente ecuación:

$$AIC = 2k - 2\ln(L)$$

Variables:

- *L*: Verosimilitud, mide qué tan bien el modelo explica los datos.
- *k*: Número de parámetros en el modelo.

3.1.5.2. Criterio de Información Bayesiano (BIC)

Adopta un enfoque más estricto respecto a la complejidad del modelo. Se basa en la estadística bayesiana, lo que añade una penalización mayor para los modelos con más parámetros, especialmente cuando el conjunto de datos es grande [34].

La cual se describe en la siguiente ecuación:

$$BIC = k\ln(n) - 2\ln(L)$$

Variables:

- *n*: Número de observaciones en el conjunto de datos.
- *k*: *número de datos*
- *L*: Verosimilitud, mide qué tan bien el modelo explica los datos.

3.1.6. HMM - MODELOS OCULTOS DE MARKOV

Un modelo oculto de Markov (o HMM *Hidden Markov Models*) es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Markov de parámetros

desconocidos. El objetivo es determinar los parámetros ocultos a partir de los parámetros observables, Los parámetros extraídos se pueden emplear para llevar a cabo sucesivos análisis [18].

Así mismo, los HMM se basan en el aumento de la cadena de Markov, es decir, una cadena de Markov es un modelo que dice algo sobre las probabilidades de secuencias de variables aleatorias y estados en los que cada uno de los cuales puede tomar valores de algún conjunto. Estos conjuntos pueden ser palabras, etiquetas o símbolos que representen cualquier cosa, como el clima. Una cadena de Markov parte de la fuerte suposición de que, si queremos predecir el futuro en la secuencia, lo único que importa es el estado actual. Los estados anteriores al estado actual no tienen impacto en el futuro excepto a través del estado actual. Es como si para predecir el tiempo de mañana se pudiera examinar el tiempo de hoy pero no se permite mirar el tiempo de ayer [19].

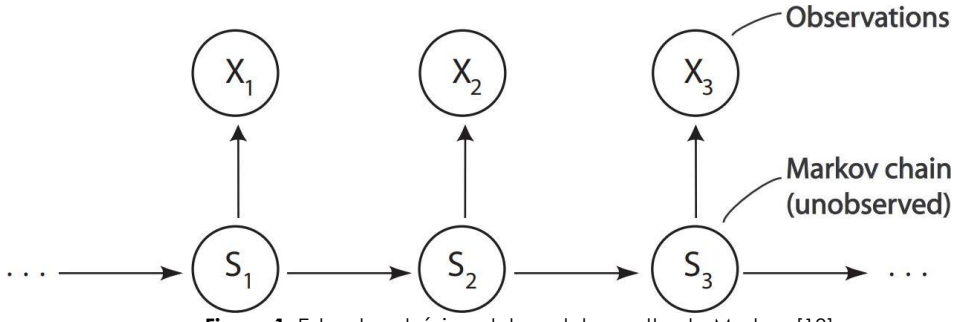

Figura 1: Estructura básica del modelo oculto de Markov [13].

Los modelos son una combinación de los dos procesos siguientes:

- Cadena de Markov St que determina el estado en el instante t, y
- Proceso dependiente del estado Xt que genera la observación dependiendo del estado actual de St

Un HMM es un doble proceso estocástico en donde el proceso que no es visible solo puede ser observado a través de otro proceso que produce una secuencia de observaciones, para lo cual se describen las ecuaciones fundamentales claves del Modelo Oculto de Markov, a continuación:

3.1.4.1. Propiedad de independencia condicional: Establece la observación actual $X_t$que depende solo del estado oculto actual S_t, y no de observaciones o estados anteriores, descrita en la siguiente ecuación [20]:

$$P(X_t = x_t \mid X_0^{t-1}, S_0^t) = P(X_t = x_t \mid S_t = s_t)$$

Variables:

- X_t: Observación (dato visible) en el tiempo t.
- x_t: Valor observado en el tiempo t.

- S_t: Estado oculto (no observable directamente) en el tiempo t.
- X_0^{t-1}: Todas las observaciones anteriores desde X_0hasta X_{t-1}.
- S_0^t: Todos los estados ocultos desde S_0 hasta S_t.

3.1.4.2. Función de Verosimilitud: Calcula la probabilidad total de observar una secuencia de datos dados los parámetros del modelo, descrita en la siguiente ecuación [20]:

$$L(\theta) = \pi P(x_0) T P(x_1) T \dots T P(x_{\tau-1}) 1^T$$

Variables:

- $L(\theta)$: Función de verosimilitud del modelo dado el conjunto de parámetros θ.
- θ: Conjunto completo de parámetros del modelo (probabilidades de transición, distribución inicial y probabilidades de emisión).
- π: Vector fila de la distribución inicial de los estados ocultos.
- $P(x_\tau)$: Matriz diagonal de probabilidades de observación, donde cada diagonal tiene $b_j(x_t) = P(X_t = x_t | S_t = j)$.
- T: Matriz de transición de estados ocultos.
- τ: Longitud de la secuencia temporal.
- 1^T: Vector columna de unos, utilizado para completar el producto matricial y obtener un escalar.

3.1.4.3. Distribución de mezcla de observaciones: Calcula la probabilidad total de observar un dato, combinando las probabilidades en todos los estados ocultos, descrita en la siguiente ecuación [20]:

$$f(x_t) = \sum_{j=0}^{J-1} \pi j \, f_j(x_t)$$

Variables:

- $f(x_t)$: Densidad marginal de la observación x_t.
- πj: probabilidad inicial de estar en el estado oculto j.
- $f_j(x_t)$: Función de densidad o probabilidad de observar x_t dado que el estado es j
- J: Número total de estados ocultos.

3.1.4.4. Matriz de transición: Define las probabilidades de pasar de un estado oculto a otro, descrita en la siguiente ecuación [20]:

$$T = \begin{bmatrix} p00 & \cdots & p0, J-1 \\ \vdots & \ddots & \vdots \\ pj-1,0 & \cdots & pj-1, J-1 \end{bmatrix}$$

Variables:

- p_{ij}: Probabilidad de transición del estado oculto i al estado j.
- J: Número total de estados ocultos.

3.1.4.5. Distribución estacionaria: Calcula la distribución de estados en equilibrio, es decir, las probabilidades de estar en cada estado oculto cuando el sistema se ha estabilizado, descrita en la siguiente ecuación [20]:

$$\pi_s = \pi_s T \; con \sum_i \pi_{s_i} = 1$$

Variables:

- π_s: Vector de distribución estacionaria (probabilidad de estar en cada estado en equilibrio).
- T: Matriz de transición.
- π_{s_i}: Componente iii del vector estacionario.

3.1.7. MÉTRICAS DE VALIDACIÓN

Los ingenieros de datos emplean una gran cantidad de tiempo mejorando la calidad de los resultados de los modelos obtenidos, normalmente no es posible saber visualmente si un modelo es mejor que otro; resulta necesario procesar algunas medidas de calidad para conocer la exactitud de los resultados [15]. En efecto, para la pruebas y evaluaciones de los modelos desarrollados, se utilizaron el algoritmo de Viterbi, descritos en la sesión 5.5.3, el cual, determina la secuencia más probable de estados ocultos a partir de los datos observables, esta secuencia permite anticipar observaciones futuras, clasificar secuencias o encontrar patrones en los datos, y así con las métricas *F1-score* y *Accuracy*, lo cuales permiten evaluar el rendimiento del modelo comparando los estados predichos con las etiquetas realas [21], lo cuales se explican a continuación:

3.1.7.1. Métrica F1 (*F1-score*)

F1 score, evalúa que tan clasificadas están las instancias que fueron clasificadas como positivas. Se define como la media armónica de la precisión y el recall [22], calculado como:

$$F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R}$$

F1-score toma valores entre 0 y 1, en donde 1 indica un modelo perfecto, es decir, tanto la precisión como el recall son perfectos y 0 indica que el modelo no tiene capacidad predictiva.

3.1.7.2. Métrica de Exactitud (*Accuracy metrics*).

Accuracy, en español se denomina Exactitud, y representa la cantidad de instancias correctamente clasificadas con respecto al total de instancias [22], calculado como:

$$A = \frac{TP + TN}{n}$$

Accuracy toma valores entre 0 y 1, en donde 1 indica un modelo preciso y 0 indica que el modelo no está fallando en su capacidad predictiva.

3.2. ANTECEDENTES

A continuación, se detallan documentos de trabajos previos que tienen alusión al proyecto a desarrollar.

- En el ámbito nacional, se tiene como referencia la tesis de grado "ACTIVISM: Plataforma para el análisis de información sobre de flujos de texto en redes sociales", de la Maestría en Ingeniería de la Información, de la Universidad de los Andes, la cual se centra en, diseñar e implementar un sistema de información que, a partir de datos provenientes de una red social, permita analizar los contenidos asociados con temas de un contexto particular, de manera que los usuarios de la red puedan encontrar hallazgos y que puedan tomar decisiones informadas sobre los contenidos publicados, para lo cual utilizaron los tuits o publicaciones que realizaron los usuarios en la plataforma web Twitter, entrenando los modelos Support Vector Machine, Naïve Bayes, Modelos ocultos de Markov y Conditional Random Fields.

  En los resultados, se realizaron 16 experimentos variando los parámetros del pipeline, y en validación del modelo alcanza valores de 90% en precisión y 85% en recall, de la entidad de tipo dinero [23].

  En el proyecto a desarrollar se propone una clasificación de los comentarios o reseñas web, pero solo se van a implementar los Modelos ocultos de Markov con un énfasis en Análisis de Sentimientos con técnicas de procesamiento del lenguaje natural (PNL).

- El siguiente trabajo fue publicado en la conferencia internacional IEEE del año 2019, sobre Big Data, Cloud Computing, Data Science y Engineering, denominado "Modelos de Markov ocultos para el análisis de sentimientos en las redes sociales". En resumen, examinan una gran cantidad de textos cargados de opiniones de Internet y evidencian que es necesario un análisis de sentimientos eficaz, que analice, reconozca y comprenda sentimientos y emociones en las opiniones dadas, para lo cual realizaron un estudio experimental para examinar el desempeño de los Modelos

Ocultos de Markov, en donde seleccionaron unos conjuntos de datos públicos que utilizaron ampliamente para el análisis de sentimientos y finalmente evaluará el desempeño para especificar el sentimiento identificado.

En los resultados de la evaluación, indicaron que los modelos ocultos de Markov logran un rendimiento superior en comparación con los algoritmos tradicionales de aprendizaje automático y destacan que son escalables y precisos al analizar el contenido generado por el usuario y al especificar opiniones y actitudes [7].

Del mismo modo, en el proyecto se propone realizar un análisis de sentimientos con el Modelo Oculto de Markov, pero con un análisis previo de minería de texto con técnicas de procesamiento del lenguaje natural (PNL).

- En la 15ª Conferencia Internacional de Electrónica, Informática y Computación (ICECCO), se publicó un artículo denominada "Una revisión sistemática de los modelos ocultos de Markov para el análisis de sentimiento", que consta de una revisión literaria sobre la aplicación de los modelos ocultos de Markov en el campo del análisis de sentimientos en relación con un proyecto de investigación sobre representación semántica y el uso de modelos gráficos probabilísticos para la determinación del sentimiento en datos textuales.

  Además, en los resultados obtenidos sugieren que los TextHMM han mostrado resultados prometedores y que existe la posibilidad de lograr una mayor precisión, y adicionalmente en el caso del uso, en lo que otros algoritmos de aprendizaje automático pueden fallar, los modelos gráficos probabilísticos como el HMM pueden mejorar resultados.

  Finalmente, la revisión de este trabajo con el proyecto que se propone da una base literaria sobre la aplicación de los modelos ocultos de Markov para el análisis de sentimiento que se desarrollará [3].

4. PREPROCESAMIENTO DE DATOS Y TRATAMIENTO DE CADENAS DE TEXTOS

En esta sección se describe el preprocesamiento de los datos y el tratamiento de las cadenas de texto, en el cual se realizan tareas como la recolección de la base de datos, la exploración del set de datos, la selección de los datos etiquetados y la limpieza de datos como faltantes y errores tipográficos.

Por consiguiente, en la preparación de los datos, se debe llevar a cabo numerosas tareas específicas tales como la identificación de la fuente de los datos, su limpieza, la eliminación de información que está fuertemente correlacionada, la búsqueda de información sesgada, y la realización de las normalizaciones necesarias [24].

4.1. RECOLECCIÓN DE LA BASE DE DATOS

En el desarrollo del proyecto, utilizamos dos bases de datos públicas para el entrenamiento del modelo, empleando características distintas para evaluar su rendimiento. El conjunto de datos cuenta con un etiquetado de múltiples clases.

A continuación, se detalla la descripción de las bases de datos seleccionadas:

- El dataset denominado Sentiment Polarity Annotations Dataset (SPOT), es un recurso de acceso libre el cual contiene en 197 reseñas extraídas de los sitios Yelp y IMDb, específicamente del Yelp Dataset Challenge disponible en https://github.com/EdinburghNLP/spot-data/tree/master?tab=readme-ov-file.
  Este conjunto de datos incluye etiquetas de polaridad a nivel de segmento, categorizadas como positivas, neutrales y negativas. Estas anotaciones han sido recopiladas con una granulación de oraciones [7]:

4.2. EXPLORACIÓN DE LOS SET DE DATOS.

4.1.1. Datasets reseñas de Sentiment Polarity Annotations Dataset (SPOT).

Los datasets de Anotaciones de Polaridad de Sentimientos (Sentiment Polarity Annotations Dataset, SPOT) constituye un recurso de libre acceso obtenido a partir de la investigación llevada a cabo por S. Angelidis y M. Lapata, titulada "Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis" (Redes de Aprendizaje de Instancias Múltiples para el Análisis de Sentimientos Detallado), publicada en Transactions of the Association of Computational Linguistics, volumen 6, páginas 17-31, en 2018, adicionalmente el dataset está alojada en GitHub, una plataforma integral para compartir, colaborar y administrar proyectos colaborativos de desarrollo. El conjunto de datos se presenta en un archivo plano compartido por Stefanos Angelidis, investigador e ingeniero especializado en aprendizaje automático y procesamiento del lenguaje natural en la Universidad de Edimburgo.

	Yelp		**IMDb**	
Datasets	Oraciones	EDUs	Oraciones	EDUs
Segmentos	1.065	1.313	1.029	1.998

Tabla 2: Características del dataset de SPOT

Los recursos de Anotaciones de Polaridad de Sentimiento SPOT, cuenta con cuatro (4) archivos (datasets), referente al sitio Yelp, está compuesto por 100 reseñas, dividido con dos dataset, el primero con 1.065 oraciones y el segundo con 1.313 EDUs (Elementary Discourse Unit), y los otros archivos del sitio IMDb compuesto con 97 reseñas, divido en dos dataset, el primero con 1.019 oraciones y el segundo con 1.998 EDUs (Elementary Discourse Unit)

En relación al proyecto aplicado, solo se utilizaron los datasets con el contenido de las reseñas dividido en oraciones, debido a que se utilizaron técnicas de Procesamiento del Lenguaje Natural (PLN) como la tokenización descrita más adelante, ya que los EDUs (Elementary Discourse Unit) contienen las palabras de las oraciones.

La siguiente gráfica de barras se representa el dataset del sitio Yelp dividido la reseña web en oraciones la variable categórica sentimiento, la cual tiene tres valores: positivo (+) con 425 repeticiones, negativo (-) con 409 repeticiones y neutro (ø) con 231 repeticiones.

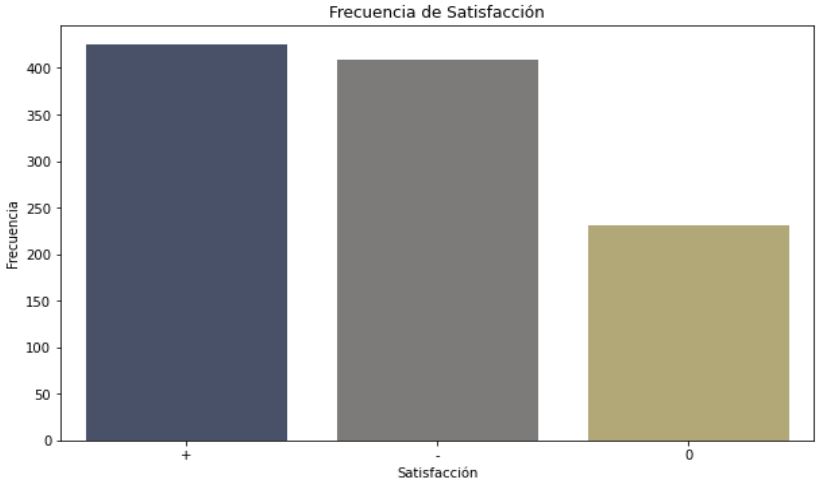

Figura 2: Distribución de Satisfacción del dataset del sitio Yelp - Oraciones.

La siguiente gráfica de barras se representa el dataset del sitio IMDb dividido la reseña web en oraciones la variable categórica sentimiento, la cual tiene tres valores: positivo (+) con 390 repeticiones, negativo (-) con 339 repeticiones y neutro (ø) con 300 repeticiones.

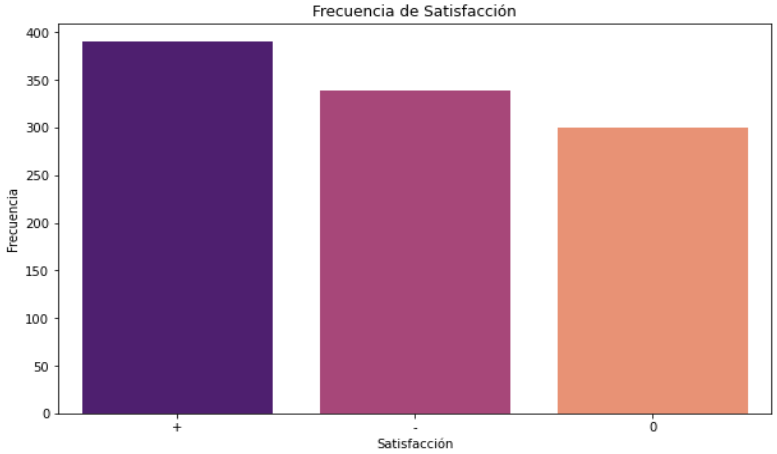

Figura 3: Distribución de Satisfacción del dataset del sitio IMDb - Oraciones.

4.3. SELECCIÓN DE LOS DATOS ETIQUETADOS.

Como se mencionó anteriormente, el set de datos denominado Sentiment Polarity Annotations Dataset (SPOT), contiene dos (2) dataset, el cual tiene una clasificación (+) positivas, (⌀) neutrales y (-) negativas, adicionalmente contiene otro tipo de clasificación para cada reseña, en el caso de los dataset del sitio Yelp, contiene una clasificación numérica de 0 a 4, el número 0 con las reseñas más negativas, hasta 4, con las reseñas más positivas, y en el caso del sitio IMDb, contiene una clasificación adicional, numérica de 0 a 9 para cada reseña, el número 0 con las reseñas más negativas, hasta 9 con las reseñas más positivas.

Por consiguiente, la decisión respecto al etiquetado es solo trabajar con las etiquetas mencionadas anteriormente (+) positivas, (⌀) neutrales y (-) negativas, que expresan con mayor exactitud el sentimiento de la reseña, para lo cual, se eliminó la clasificación adicional por medio de la librería Pandas por medio de la función: *dropna()*.

4.4. PREPARACIÓN DE LOS DATOS.

En esta etapa se realiza el preprocesamiento de los datos, en el cual se realiza limpieza de los datos como, verificación de datos faltantes, eliminación de caracteres especiales, correcciones de errores tipográficos, y transformación como, convertir los caracteres a minúsculas y eliminación de signos de puntuación, entre otros.

a. Datos Faltantes: En todos los dataset se realiza una comprobación de datos faltantes *NaN* por medio de la función *isna()*, y adicionalmente se realiza la sumatoria de cada uno de ellos, entregado como resultado cero datos faltantes, todos los dataset no contienen datos faltantes.
b. Preparación de los dataset, como: renombrar las columnas, crear una nueva columna con el nombre de sentimiento y asignarle el valor del índice que contiene la etiqueta, posteriormente restablecer los valores del índice en un rango numérico secuencial, y la eliminación de las filas que tienen valores NaN (valores nulos o faltantes) que corresponden a la clasificación adicional, de 0 a 9 para cada reseña, mencionada en el numeral 4.3.
c. Transformar caracteres a minúsculas: Con el atributo *str* de pandas se accede a las funciones vectorización de las cadenas de texto, para realizar la conversión de los caracteres a minúsculas con método *lower*.
d. Manejo de contracciones en inglés: La librería contractions en python expande automáticamente las contracciones comunes en inglés de un texto, como por ejemplo *It's amazing* a *It is amazing*, como parte de preprocesamiento de texto.
e. Reemplazo de caracteres tipográficos: Los signos (-), (/) y (\) conocidos como barra y barras invertidas, utilizados para unir palabras son reemplazados por un espacio en blanco para separarlas por medio de una función.
f. Corrección de errores ortográficos: Apoyados en la biblioteca *pyspellchecker*, se procede a realizar la identificación y corrección de los errores ortográficos.
g. Eliminación de signos de puntuación: En la eliminación de los signos de puntuación como; los puntos (.), comas (,), signos de exclamación (!), signos de interrogación (?), dos puntos (:), punto y coma (;), comillas simples ('), comillas dobles ("), paréntesis (()

y)), corchetes ([) y (]), llaves ({) y (}), guiones (-), guiones bajos (_), barras (/ y), signos de arroba (@), signos de número (#), signos de dólar ($), porcentajes (%), signos de ampersand (&), asteriscos (*), signos más (+), signos igual (=), signos mayor que (>) y signos menor que (<), se crea una función para su eliminación con la librería *re,* ya que proporciona soporte para operaciones con expresiones regulares.

h. Estandarización: Como medida opcional y como parte de las pruebas a evaluar, se estandarizó las matrices vectoriales para analizar los resultados con un conjunto de datos transformados, ya que podrían comportarse mal si las características individuales no se parecen más o menos a los datos distribuidos normalmente estándar, de manera que se utilizó la clase *StandardScaler* de la biblioteca *Scikit-learn*, la que estandariza las características eliminando la media y escalando a la varianza unitaria, donde la puntuación estándar de una muestra *X* se calcula como:

$$Z = \frac{X - \mu}{\sigma}$$

Donde, *X* es el valor original de la variable, μ es la media de la variable y σ: es la desviación estándar, este cálculo transforma los datos para que tengan una media igual a cero y una desviación estándar igual a 1[25].

4.5. PROCESAMIENTO DEL LENGUAJE NATURAL PNL.

NLTK (Natural Language Toolkit) es una biblioteca de Python para el procesamiento de lenguaje natural, que ofrece herramientas para realizar tareas de NLP, como en este caso la tokenización, eliminación de stopwords, lematización, y clasificación de palabras.

4.1.2. Tokenización

En esta instancia se emplea el paquete *punkt* de *nltk*, que es necesario para el tokenizador de palabras y el resultado es guardado en una nueva variable *Texto_Token*.

Finalmente, los resultados son llevados a una figura con una nube de palabras, apoyados de las bibliotecas *pandas*, *matplotlib* y *wordcloud* de la variable Texto_Token y adicionalmente se calcula la frecuencia de cada palabra, apoyados de la clase *Counter* del módulo *collections*.

Figura 4: Nube de Palabras de la variable *Texto_Token* del sitio Yelp Oraciones.

Palabra	Frecuencia
the	705
i	490
and	443
a	358
to	332
was	308
it	256
is	254
not	199
of	176
was	308
it	256
is	254
not	199
of	176
i	490
is	254
not	199
of	176
in	153
for	149
we	138
that	137
my	119

Tabla 3: Frecuencia de Palabras de la variable *Texto_Token* del sitio Yelp Oraciones

4.1.3. Eliminación de stopwords.

Las palabras vacías son palabras en cualquier idioma o corpus que aparecen con frecuencia, pero para algunas tareas de PNL, no proporcionan ninguna información adicional o valiosa al texto que las contiene, palabras como un, ellos, el, es, un, etc [26], de manera que se utilizó la biblioteca de NLTK para eliminar la lista predeterminada de palabras de la biblioteca.

4.1.4. Lematización.

El objetivo principal de la lematización es normalizar las palabras para facilitar su análisis y comprensión. Al reducir las palabras a su forma base, se pueden identificar más fácilmente las palabras relacionadas y tratarlas como variantes de una misma palabra, lo que simplifica tareas como la búsqueda, clasificación y extracción de información en texto [27], de modo que se utilizó la herramienta del módulo nltk.stem.wordnet, que se apoya en la base de datos léxica de WordNet.

4.1.5. Clasificación de palabras.

Por último, de las técnicas de procesamiento de lenguaje natural, se realizó la clasificación de palabras con la finalidad de incluir un nuevo dataset en el conjunto de pruebas, para la clasificación se desarrolló con el etiquetado POS, etiquetado de partes de discurso, lo cual es un proceso para marcar las palabras en formato de texto para una parte particular de un discurso según su definición y contexto [28], en los resultados obtenidos del etiquetado

se obtienen a nivel general la siguiente abreviatura de cada palabra:

Abreviatura	Significado
CC	Conjunción de Coordinación
CD	Dígito Cardinal
DT	Determinante
EX	Existencial Allí
FW	Palabra Extranjera
IN	Preposición/Conjunción Subordinante
JJ	Esta Etiqueta NLTK POS es un Adjetivo (Grande)
JJR	Adjetivo, Comparativo (Más Grande)
JJS	Adjetivo, Superlativo (Más Grande)
LS	Lista De Mercado
MD	Modal (Podría, Voluntad)
NN	Sustantivo, Singular (Gato, Árbol)
NNS	Sustantivo Plural (Escritorios)
PNN	Nombre Propio, Singular (Sarah)
NNPS	Nombre Propio, Plural (indios O americanos)
PDT	Predeterminador (Todos, Ambos, La Mitad)
TPV	Terminación Posesiva (Padre\ 'S)
PRP	Pronombre Personal (Ella, Ella, Él, Él Mismo)
PPR	Pronombre Posesivo (Ella, Su, Mío, Mi, Nuestro)
RB	Adverbio (Ocasionalmente, Rápidamente)
RBR	Adverbio, Comparativo (Mayor)
RBS	Adverbio, Superlativo (Más Grande)
RP	Partícula (Sobre)
A	Marcador Infinito (A)
UH	Interjección (Adiós)
VB	Verbo (Preguntar)
JBV	Verbo Gerundio (Juzgar)
VBD	Verbo En Pasado (Suplicó)
VBN	Verbo Participio Pasado (Reunificado)
VBP	Verbo, Tiempo Presente, No Tercera Persona del Singular (Envoltura)
VBZ	Verbo, Tiempo Presente Con 3ª Persona del Singular (Bases)
WDT	Wh-Determinante (Eso, Qué)
WP	Wh- Pronombre (Quién)
WRB	Wh- Adverbio (Cómo)

Tabla 4: Resultado etiquetado POS [28].

5. MODELADO A PARTIR DE MODELOS PROBABILÍSTICOS DE DATOS ASOCIADOS CON RESEÑAS

El modelado se suele ejecutar en múltiples iteraciones, normalmente, los analistas de datos ejecutan varios modelos utilizando los parámetros predeterminados y ajustan los parámetros o vuelven a la fase de preparación de datos para las manipulaciones necesarias del modelo [27].

En el desarrollo del proyecto se empleó un modelo probabilístico o estadístico, en este caso el Modelo Oculto de Márkov, el cual costa de las técnicas de extracción de características del entrenamiento del modelo propuesto, la implementación del modelo en Python, la estimación de parámetros para el modelo y la ejecución del modelo.

5.1. TÉCNICAS DE EXTRACCIÓN DE CARACTERÍSTICAS

Para comenzar, en el modelado se utilizaron técnicas de extracción, que permite mejorar el rendimiento del modelo, dado que es un proceso fundamental en el análisis de datos textuales, que consisten en convertir el texto en datos numéricos que puedan ser procesados por algoritmos y para el modelo propuesto se utilizaron tres enfoques, el Bag of Words, TF-IDF y Embeddings.

- **Bag of Words (BoW)**: Es un enfoque comúnmente utilizado en la extracción de características en texto que consiste en crear una bolsa o conjunto de todas las palabras únicas presentes en un corpus [29], para lo cual se utilizó módulo ***CountVectorizer*** de ***scikit-learn*** para convertir el texto preprocesado en una representación de Bag of Words (vector de frecuencias de palabras), para lo cual se limitó el vocabulario a las 5000 palabras más frecuentes, con el parámetro:
  - *max_features=5000*

- **TF-IDF (Term Frequency - Inverse Document Frequency)**: Es un enfoque que combina la frecuencia de términos (TF) y la frecuencia inversa de documentos (IDF) para asignar pesos a las palabras en el corpus [29], para lo cual se utilizó módulo ***TfidfVectorizer*** de ***scikit-learn*** que convierte el texto en vectores de TF-IDF, donde se ponderan las palabras en función de su importancia en el documento y su ocurrencia global en todo el corpus y se limita el vocabulario a las 5000 palabras más informativas, con el parámetro:
  - *max_features=5000*

- **Word Embeddings**: Son vectores densos y de baja dimensión que representan palabras de manera que palabras con significados similares tengan representaciones vectoriales cercanas [30], para lo cual se utilizó el módulo ***Word2Vec*** de la biblioteca ***gensim***, que generar embeddings de palabras basado en la secuencia de las palabras en el texto, con los siguientes parámetros:
  - *vector_size=50*: Un vector de 50 dimensiones para cada palabra.
  - window=5: Consideración de 5 palabras a la izquierda y 5 palabras a la derecha.
  - min_count=1: Inclusión de palabras para que aparezcan al menos una vez.

- workers=4: Definición del número de núcleos de CPU a usar para el entrenamiento.

- **GloVe 300d**: Es un modelo de representaciones vectoriales de palabras (word embeddings) entrenado con el algoritmo GloVe (Global Vectors) de Stanford, que genera vectores de 300 dimensiones para cada palabra basándose en la matriz global de coocurrencias de un corpus masivo de texto; captura similitudes semánticas y relaciones lineales (como analogías), y es ampliamente usado en tareas de procesamiento de lenguaje natural como clasificación, análisis de sentimientos o inicialización de capas de embedding en modelos de redes neuronales.

- **Word2Vec-384d**: Es un modelo de vectores de palabras que representa cada término en un espacio de 384 dimensiones, entrenado para predecir palabras en su contexto cercano dentro de grandes corpus de texto; este enfoque captura relaciones semánticas y sintácticas de forma eficiente, permitiendo.
  - vector_size=384: define la cantidad de números que usará el modelo para representar cada palabra.
  - window=5: Consideración de 5 palabras a la izquierda y 5 palabras a la derecha.
  - min_count=1: Inclusión de palabras para que aparezcan al menos una vez.
  - workers=4: Definición del número de núcleos de CPU a usar para el entrenamiento.

- **FastText-300**: Es un modelo de embeddings de palabras de 300 dimensiones, que extiende Word2Vec incorporando sub-palabras (n-gramas de caracteres), lo que permite manejar mejor palabras raras o desconocidas y capturar similitudes morfológicas y lingüísticas, especialmente útil en lenguajes ricos en flexión.
  - *vector_size=300*: Un vector de 300 dimensiones para cada palabra.
  - window=5: Consideración de 5 palabras a la izquierda y 5 palabras a la derecha.
  - min_count=1: Inclusión de palabras para que aparezcan al menos una vez.
  - workers=4: Definición del número de núcleos de CPU a usar para el entrenamiento.

- **BERT-base**: Es un modelo de lenguaje basado en la arquitectura de transformadores, con 12 capas (transformer blocks) y 768 dimensiones de embedding por token; se entrena con atención bidireccional sobre el contexto completo de cada palabra, produciendo representaciones dinámicas y dependientes del contexto, lo que lo hace particularmente poderoso en tareas como comprensión de texto, preguntas y respuestas, o clasificación de secuencias.
  - return_tensors='pt': Devuelve tensores de PyTorch.
  - truncation=True: Corta el texto si excede 512 tokens
  - max_length=512: Límite máximo para entrada (tokens BERT)
  - padding=True: Agrega padding automático para igualar longitud

5.2. ESTIMACIÓN DE PARÁMETROS PARA EL MODELO

Por otro lado, para el modelamiento y mejorar el rendimiento del modelo de Modelo Oculto de Markov se seleccionan los parámetros de la función *GaussianHMM*, para mejorar el rendimiento del Modelo Oculto de Markov, seleccionan los parámetros de la función, descritos a continuación:

- **Número de Estados Ocultos** (*n_components*): Se seleccionaron a través de experimentos y validaciones usando métricas de BIC (Bayesian Information Criterion) y AIC (Akaike Information Criterion) para determinar el número óptimo de estados ocultos, los cuales se determinaron en un rango de 2 a 52, validado con los siguientes hiper-parámetros:
    - Tipo de Covarianza, el parámetro controla cómo se modelan las covarianzas de las distribuciones gaussianas en los estados ocultos, y se define como *covariance_type*, las matrices de covarianza seleccionadas para los experimentos son:
        - **Diagonal (*diag*)**: Cada estado utiliza una matriz de covarianza diagonal.
        - **Completa (*full*)**: Cada estado utiliza una matriz de covarianza completa.
        - **Esférica (*spherical*)**: Cada estado utiliza un único valor de varianza que se aplica a todas las características.
        - **Compartida o Vinculada (tied):** Todos los estados utilizan la misma matriz de covarianza completa.
    - Número de Iteraciones, parámetro que establece el número máximo de iteraciones del algoritmo de *Baum-Welch*, definido como *n_iter*, el cual se fijó en 100.
    - Semilla Aleatoria (random_state), la cual es útil para reproducir resultados, se fija con un valor de 42 para que los resultados sean reproducibles.

5.3. SELECCIÓN DEL NÚMERO DE ESTADOS OCULTOS CON AIC/BIC.

En continuación con la estimación de parámetros para el modelo, se realizaron experimentos con las variaciones de los parámetros como las matrices de vectorización, el tipo de la covarianza, y la clasificación POS Tagging, para determinar el número de estados ocultos, según el Criterio de Información Bayesiano (BIC) y el Criterio de Información de Akaike (AIC).

5.3.1. Selección del número de estados ocultos para el dataset IMDb.

En la siguiente tabla se muestran los resultados para diferentes configuraciones de vectorizaciones (BoW, TF-IDF, Embedding, GloVe, Word2Vec) y parámetros de covarianza (diag, full, tied, spherical) con y sin POS Tagging o normalización, para los criterios de Información Bayesiano (BIC) y los criterios de Información de Akaike (AIC) en función del número de estados ocultos.

Figura	Vectorización	Tipo de Covarianza	Normalización de Datos	POS Tagging	N. Estados

					BIC	AIC
8	Bag of Words	diag	Si	Si	17	17
9	Bag of Words	full	Si	Si	35	35
10	Bag of Words	tied	Si	Si	50	50
11	Bag of Words	spherical	Si	Si	10	50
12	Bag of Words	diag	No	Si	8	8
13	Bag of Words	full	No	Si	40	40
14	Bag of Words	tied	No	Si	40	40
15	Bag of Words	spherical	No	Si	12	40
16	TF-IDF	diag	Si	Si	4	4
17	TF-IDF	full	Si	Si	40	40
18	TF-IDF	tied	Si	Si	50	50
19	TF-IDF	spherical	Si	Si	40	50
20	TF-IDF	diag	No	Si	50	50
21	TF-IDF	full	No	Si	10	10
22	TF-IDF	tied	No	Si	40	50
23	TF-IDF	spherical	No	Si	20	40
24	Embedding	diag	Si	Si	40	50
25	Embedding	full	Si	Si	3	3
26	Embedding	tied	Si	Si	20	50
27	Embedding	spherical	Si	Si	10	50
28	Embedding	diag	No	Si	2	2
29	Embedding	full	No	Si	4	4
30	Embedding	tied	No	Si	3	3
31	Embedding	spherical	No	Si	3	3
32	Bag of Words	diag	Si	No	4	4
33	Bag of Words	full	Si	No	4	4
34	Bag of Words	tied	Si	No	4	4
35	Bag of Words	spherical	Si	No	4	4
36	Bag of Words	diag	No	No	4	4
37	Bag of Words	spherical	No	No	4	4
38	TF-IDF	diag	Si	No	4	4
39	TF-IDF	full	Si	No	4	4
40	Glove-300d	diag	No	Si	0	0
41	Glove-300d	full	No	Si	0	0
42	Glove-300d	tied	No	Si	1	1
43	Glove-300d	spherical	No	Si	2	2
44	Glove-300d	diag	Si	Si	0	0

45	Glove-300d	full	Si	Si	0	0
46	Glove-300d	tied	Si	Si	1	1
47	Glove-300d	spherical	Si	Si	1	1
48	Glove-300d	diag	No	No	6	18
49	Glove-300d	full	No	No	47	47
50	Glove-300d	tied	No	No	1	1
51	Glove-300d	spherical	No	No	7	16
52	Glove-300d	diag	Si	No	5	48
53	Glove-300d	full	Si	No	48	48
54	Glove-300d	tied	Si	No	1	1
55	Glove-300d	spherical	Si	No	4	44
56	Word2Vec-384d	diag	No	Si	11	13
57	Word2Vec-384d	tied	No	Si	2	1
58	Word2Vec-384d	spherical	No	Si	12	17
59	Word2Vec-384d	diag	Si	Si	50	50
60	Word2Vec-384d	full	Si	Si	6	5
61	Word2Vec-384d	tied	Si	Si	3	4
62	Word2Vec-384d	spherical	Si	Si	49	30

Tabla 5: Resultados de los criterios BIC y AIC del dataset IMDb

En la tabla anterior se determinó el número de estados ocultos para los criterios BIC y AIC, de acuerdo con las variaciones propuestas, en el que se observa diferentes resultados para cada experimento, por otro lado, aunque existe una similitud entre los resultados de los criterios BIC y AIC, también se observa diferencia entre ellos, debido a que ambos criterios penalizan la complejidad del modelo de forma diferente, el AIC tiende a preferir modelos más complejos y el BIC penaliza con mayor fuerza la complejidad del modelo [38].

Equivalentemente, en las gráficas a continuación, se presentan la tendencia en la selección del hiper-parámetro n_components, en el eje X número de estados ocultos y en el eje Y los criterios BIC de color azul y el AIC en color naranja, del dataset IMDb, combinando distintas configuraciones de representación del texto y estructuras de covarianza.

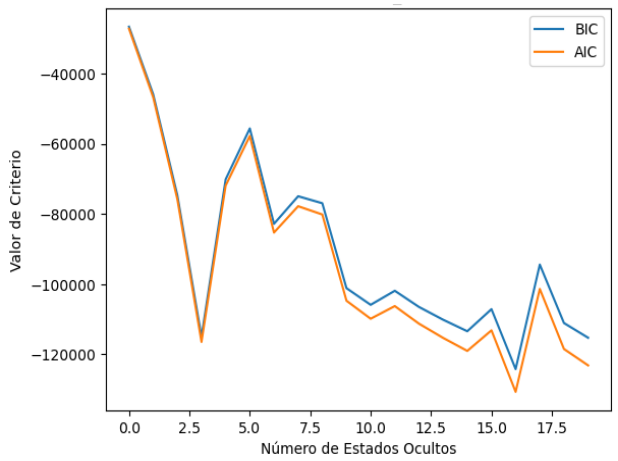

Figura N. 5: BoW, diag, normalizado y POS Tagging

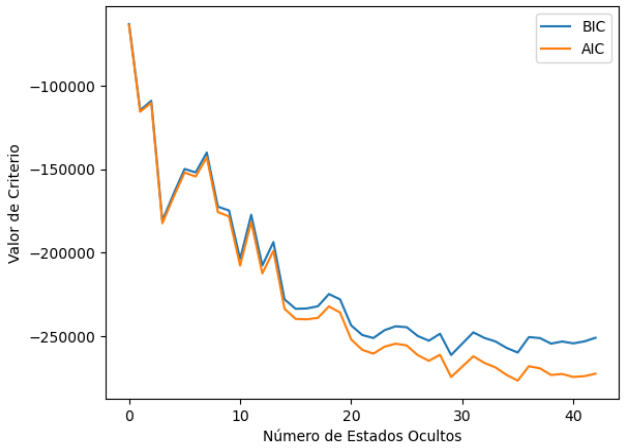

Figura N. 6: TF-IDF, full, normalizado y POS Tagging

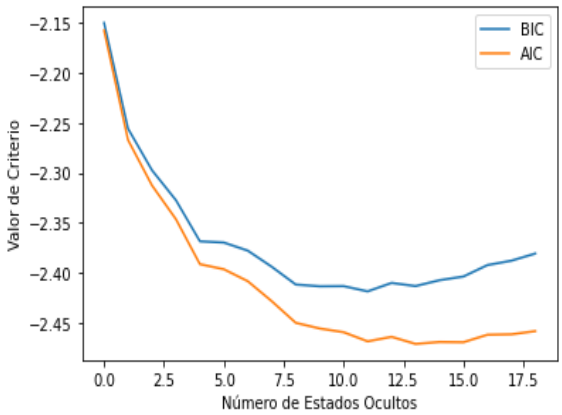

Figura N. 7: Glove-300d, sphl y Lemat_Plano-Esc

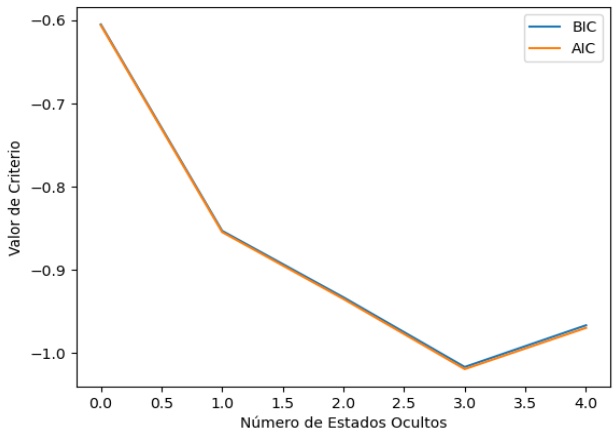

Figura N. 8: Embedding, full, normalizado y POS Tagging

En un análisis general de los gráficos de líneas del *dataset IMDb*, se observa que los dos criterios BIC y AIC tienden a disminuir conforme aumenta el número de estados ocultos, lo que indica un mejor ajuste en el modelo, aunque, a partir de cierto número de estados ocultos, las curvas presentan divergencia, adicionalmente, el criterio AIC presenta valores menores que BIC, debido a que penaliza la complejidad del modelo de forma menos estricta, pero el criterio BIC tiende a estabilizarse antes que el AIC, lo cual es esperable dado a que es más conservador frente a la complejidad y en múltiples configuraciones, ambas tendencias descienden en las iteraciones 2 a 10 estados, seguido de estabilización para los siguientes estados ocultos.

En referencia a la vectorización, los estados ocultos óptimos para BoW y TF-IDF, los mínimos criterios suelen encontrarse entre 8 a 18 estados ocultos, sugiriendo una estructura relativamente simple para capturar la dinámica de las secuencias. En las vectorizaciones Embedding y Word2Vec-384d los rangos mínimos son más amplios, con valores entre 10 a 25 estados ocultos, debido a que la dimensionalidad es mayor, y, por último, en el caso de la vectorización GloVe-300d, la tendencia presenta mayor irregularidad y en algunos casos no convergen completamente a los 50 estados ocultos, manifestando un posible sobreajuste.

5.3.2. Selección del número de estados ocultos para el dataset Yelp.

De la misma manera, se muestran los resultados de los criterios de Información Bayesiano (BIC) y los criterios de Información de Akaike (AIC).

Figura	Vectorización	Tipo de Covarianza	Normalización de Datos	Clasificación POS Tagging	N. Estados	
					BIC	AIC
63	BoW	diag	Si	Si	21	21
64	BoW	spherical	Si	Si	30	49
65	BoW	diag	Si	No	39	39
66	BoW	spherical	Si	No	9	9
67	BoW	spherical	No	Si	11	33
68	BoW	diag	No	Si	8	8
69	BoW	spherical	No	No	2	3
70	BoW	diag	No	No	10	10
71	TF-IDF	diag	Si	Si	18	18
72	TF-IDF	spherical	Si	Si	28	28
73	TF-IDF	diag	Si	No	2	2
74	TF-IDF	spherical	Si	No	5	5
75	TF-IDF	spherical	No	Si	30	35
76	TF-IDF	diag	No	Si	3	3
77	TF-IDF	spherical	No	No	1	1
78	TF-IDF	diag	No	No	6	6
79	Embedding	diag	Si	Si	26	36
80	Embedding	spherical	Si	Si	6	8
81	Embedding	diag	Si	No	4	5
82	Embedding	spherical	Si	No	3	3
83	Embedding	spherical	No	Si	5	5
84	Embedding	diag	No	Si	4	4
85	Embedding	spherical	No	No	3	4
86	Embedding	diag	No	No	4	5
87	Word2Vec-384d	diag	Si	Si	3	3
88	Word2Vec-384d	spherical	Si	Si	4	4
89	Word2Vec-384d	diag	Si	No	3	4
90	Word2Vec-384d	spherical	Si	No	4	4
91	Word2Vec-384d	spherical	No	Si	3	3
92	Word2Vec-384d	diag	No	Si	5	5
93	Word2Vec-384d	spherical	No	No	4	4
94	Word2Vec-384d	diag	No	No	3	3

95	GloVe 300d	spherical	Si	No	3	4
96	GloVe 300d	spherical	No	No	4	5
97	GloVe 300d	diag	No	No	3	3
98	GloVe 300d	diag	Si	No	4	4
99	GloVe 300d	spherical	Si	Si	4	4
100	GloVe 300d	spherical	No	Si	5	5
101	GloVe 300d	diag	No	Si	3	3
102	GloVe 300d	diag	Si	Si	4	4
103	FastText-300	diag	Si	Si	3	3
104	FastText-300	spherical	Si	Si	5	5
105	FastText-300	diag	Si	No	4	4
106	FastText-300	spherical	Si	No	3	3
107	FastText-300	spherical	No	Si	4	4
108	FastText-300	diag	No	Si	3	4
109	FastText-300	diag	No	No	3	4
110	FastText-300	spherical	No	No	5	5

Tabla 6: Resultados de los criterios BIC y AIC del dataset Yelp

De la misma manera, se observan los resultados para el dataset Yelp similares al dataset IMDb, y comparando el valor más repetido para el criterio BIC, en el dataset IMDb es de 4, igual para el dataset Yelp y en el caso del criterio AIC, el estado más repetido es de 3 para el dataset IMDb y 4 para el dataset Yelp, lo que demuestra su similitud.

Asimismo, en las gráficas a continuación, se presentan la tendencia en la selección del hiper-parámetro n_components, en el eje X número de estados ocultos y en el eje Y los criterios BIC de color azul y el AIC en color naranja, del dataset Yelp.

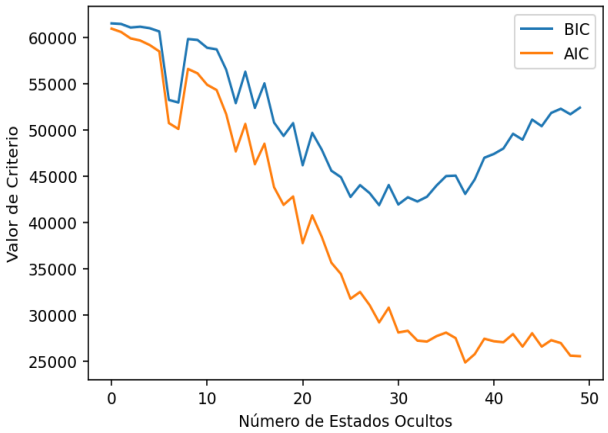

Figura N. 9: BoW, spherical y POS Tagging-Esc

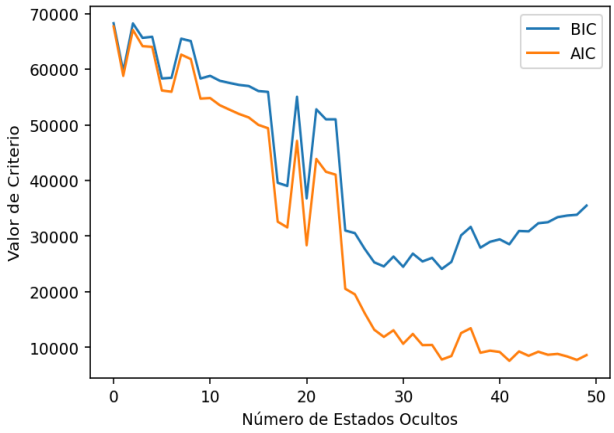

Figura N. 10: TF-IDF, Sph y POS Tagging- Esc

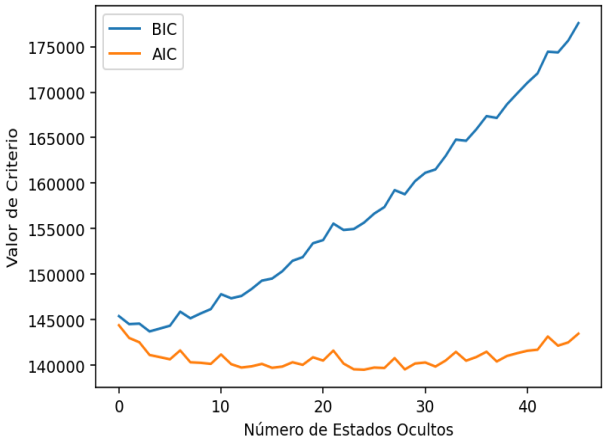

Figura N. 11: Embedding, Diag y Texto Lemat -Esc

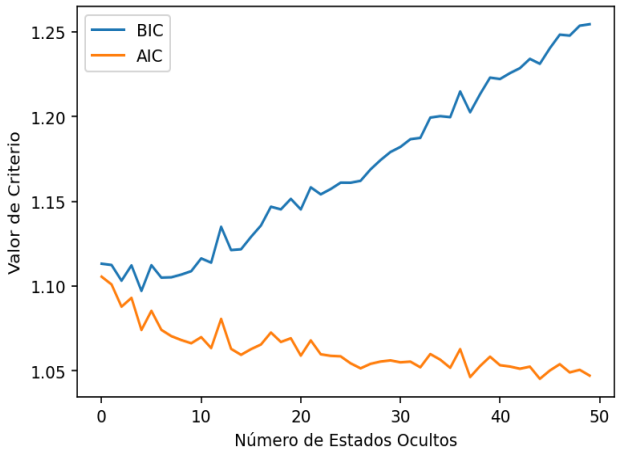

Figura N. 12: Word2Vec-384d, Sph y Texto Lema – Esc

De igual modo, el análisis general de los gráficos de líneas del *dataset Yelp*, los dos criterios AIC y BIC presentan una disminución en los primeros números de estados ocultos, seguido de una estabilización que demuestra equilibro del ajuste, el AIC presenta valores menores que el BIC, debido a que la penalización es menor.

Sin embargo, en el caso de las vectorizaciones, se observa que en algunos casos, con Embeddings, Word2Vec y FastText, comienzan a divergir a partir de los 30 a 35 estados ocultos, no obstante, los valores AIC y BIC se estabilizan con BoW y TF-IDF entre 8 y 15 estados ocultos, en el caso de Embeddings con POS Tagging y escalado en los estados ocultos entre 10 y 20, la tendencia es más estable, para la vectorización Word2Vec-384d se aprecia una mejoría gradual en la tendencia en el rango de 15 a 20 estados ocultos, en el caso de la vectorización GloVe-300d, los estados ocultos con mayor estabilidad se sitúa entre 12 a 20, y finalmente, en la vectorización FastText-300, ocurre en los estados ocultos 10 a 18, con divergencia a partir de los 25 estados.

En cuanto a la divergencia se observa en estados ocultos mayores a 30, pero específicamente para vectorizaciones Word2Vec y GloVe, para los dos criterios AIC y BIC.

Por último, a nivel general, estados ocultos con menos de 8 estados resultan sin ningún cambio en la tendencia y mayores a 30 estados muestran sobreajuste, por lo cual el rango optimo de número de estados ocultos se encuentra entre 10 a 20 estados.

5.4. ENTRENAMIENTO DEL MODELO PROPUESTO

5.4.1. Mapeo de sentimientos.

Antes de comenzar con el entrenamiento del modelo, se inició convirtiendo las etiquetas del sentimiento a valores numéricos, para lo cual creo un directorio con el mapeo de sentimientos, y así convertir las etiquetas con los símbolos a valores numéricos, de la siguiente manera:

- El signo (+) se reemplaza con el número (0)

- El signo (0) se reemplaza con el número (1)
- El signo (-) se reemplaza con el número (2)

Por último, se almacena el resultado en la variable (y), con las etiquetas para la validación en un vector.

5.4.2. División del conjunto de datos en entrenamiento y validación.

Asimismo, se dividió los conjuntos de datos, resultado de las matrices generadas de la vectorización Bag of Words, TF-IDF y Embeddings, en entrenamiento y validación con la función train_test_split de la librería scikit-learn, creado las siguientes variables:

- X_train: Datos para el entrenamiento.
- X_val: Datos para la validación.
- y_train: Etiquetas para el entrenamiento.
- y_val: Etiquetas para la validación.

Los parámetros fijados fueron, tamaño de los datos de prueba (test_size) en 10%, dejando 90% para el entrenamiento y estableciendo la semilla aleatoria (random_state) en 42.

5.4.3. Definición de Modelo Oculto de Markov (HMM).

El modelo oculto de Markov se definió con emisiones gaussianas, el cual es un tipo de HMM en donde los estados generan una secuencia de valores con distribuciones gaussianas, y para lo cual se definió el modelo establecido por la librería hmmlearn, en el que se importó el módulo completo de hmm, haciendo uso de la clase GaussianHMM para desarrollar el modelo oculto de Markov con observaciones gaussianas multivariadas [31].

5.5. IMPLEMENTACIÓN DEL MODELO EN PYTHON

Inmediatamente, después de ejecutadas las técnicas de extracción y los textos han sido transformadas en representaciones de Bag of Words, TF-IDF y Word Embeddings, además, se adicionaron cuatro vectorizaciones adicionales, GloVe 300d, Word2Vec-384d, FastText-300 y BERT-base, y se procedió con el entrenamiento y predicción del modelo.

5.5.1. Arquitectura de NLP y HMM.

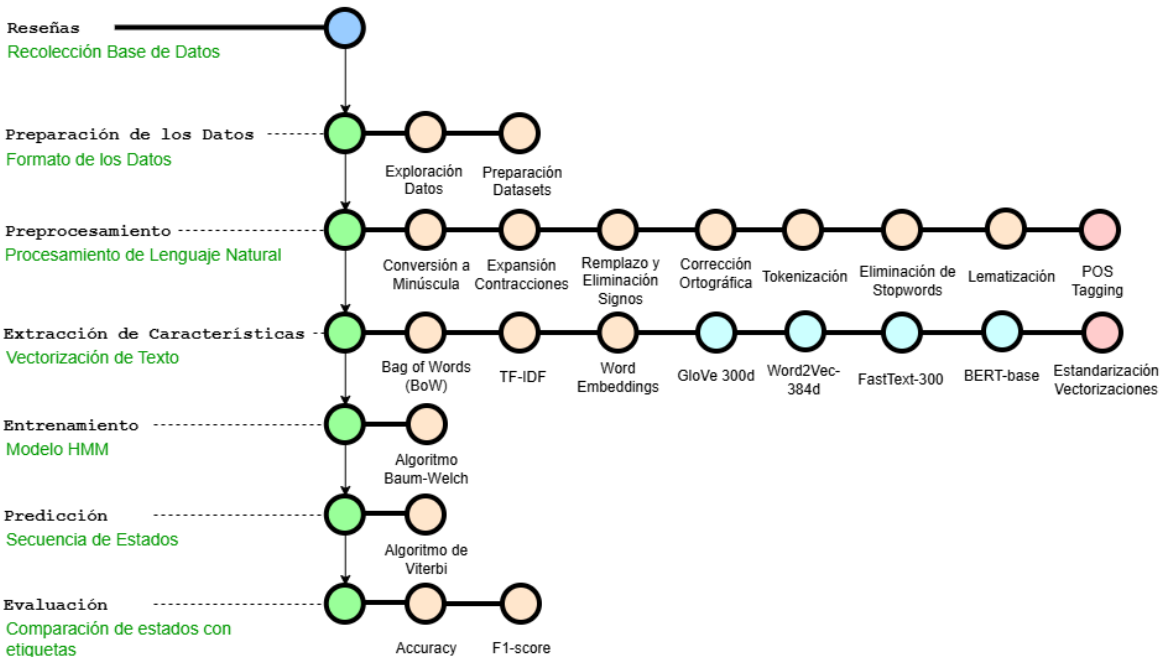

Figura N.63: Arquitectura módulo en Python

5.5.2. Entrenamiento del modelo con el algoritmo de Baum-Welch.

En el entrenamiento el Modelo Oculto de Markov gaussiano, se utilizó el algoritmo Baum-Welch, también conocido como algoritmo de expectativa-maximización (EM), el cual es un método utilizado en ciencia de datos, para refinar iterativamente las estimaciones iniciales de los parámetros del HMM, dando como resultado un modelo localmente óptimo a través del cálculo de varios valores de probabilidad [32], y se aplica a través de la siguiente línea en Python:

model.fit(X_train)

El modelo fue definido como GaussianHMM con los parámetros definidos y descrito anteriormente en la sección 5.2. (estimación de parámetros para el modelo), el cual entrena el conjunto de entrenamiento, para lo cual se utilizaron las siguientes matrices descritas en la siguiente tabla.

Nombre	Vectorización	Escalado	POS Tagging
BoW Normal	Bag of Words		
TF-IDF Normal	TF-IDF		
Embeddings Normal	Word Embeddings		
BoW Escalado	Bag of Words	✓	
TF-IDF Escalado	TF-IDF	✓	
Embeddings Escalado	Word Embeddings	✓	
BoW POS Tagging	Bag of Words		✓
TF-IDF POS Tagging	TF-IDF		✓
Embeddings POS Tagging	Word Embeddings		✓

BoW Escalado POS Tagging	Bag of Words	✓	✓
TF-IDF Escalado POS Tagging	TF-IDF	✓	✓
Embeddings Escalado POS Tagging	Word Embeddings	✓	✓
Word2Vec-384d Escalado POS Tagging	diag	✓	✓
Word2Vec-384d Escalado POS Tagging	spherical	✓	✓
Word2Vec-384d Escalado	diag	✓	
Word2Vec-384d Escalado	spherical	✓	
Word2Vec-384d POS Tagging	spherical		✓
Word2Vec-384d POS Tagging	diag		✓
Word2Vec-384d Normal	spherical		
Word2Vec-384d Normal	diag		
GloVe 300d POS Tagging	tied		✓
GloVe 300d Escalado POS Tagging	tied	✓	✓
GloVe 300d Escalado	spherical	✓	
GloVe 300d Normal	Spherical		
GloVe 300d Normal	diag		
GloVe 300d Escalado	diag	✓	
GloVe 300d Normal	tied		
GloVe 300d Normal	full		
GloVe 300d Escalado	full	✓	
GloVe 300d Escalado	tied	✓	
GloVe 300d Escalado POS Tagging	spherical	✓	✓
GloVe 300d POS Tagging	spherical		✓
GloVe 300d POS Tagging	diag		✓
GloVe 300d POS Tagging	full		✓
GloVe 300d Escalado POS Tagging	diag	✓	✓
GloVe 300d Escalado POS Tagging	full	✓	✓
FastText-300 Escalado POS Tagging	diag	✓	✓
FastText-300 Escalado POS Tagging	Spherical	✓	✓
FastText-300 Normal	full		
FastText-300 Escalado	diag	✓	
FastText-300 Normal	tied		
FastText-300 Escalado	spherical	✓	
FastText-300 POS Tagging	full		✓
FastText-300 Escalado POS Tagging	full	✓	✓
FastText-300 Escalado	full	✓	
FastText-300 POS Tagging	tied		✓
FastText-300 Escaldo POS Tagging	tied	✓	✓
FastText-300 POS Tagging	spherical		✓
FastText-300 Normal	spherical		
FastText-300 POS Tagging	diag		✓
FastText-300 Normal	diag		
BERT-base Escalado POS Tagging	diag	✓	✓
BERT-base Escalado POS Tagging	spherical	✓	✓
BERT-base Escalado	diag	✓	

BERT-base Escalado	spherical	✓	
BERT-base POS Tagging	spherical		✓
BERT-base Normal	spherical		
BERT-base POS Tagging	diag		✓
BERT-base Normal	diag		

Tabla 7: Configuración parámetros para el conjunto de entrenamiento

5.5.3. Predicción de los Estados Ocultos en el conjunto de validación con el algoritmo de Viterbi

Después de entrenar el modelo con el algoritmo Baum-Welch, se utilizó el algoritmo Viterbi el cual es un algoritmo de programación dinámica para encontrar la secuencia de estados ocultos más probables en el conjunto de validación, debido a que el algoritmo calcula la probabilidad de cada posible secuencia de estados ocultos que podría haber producido una secuencia dadas las observaciones[33], y se ejecuta mediante la siguiente línea en Python:

```
y_pred_val = model.predict(X_val)
```

En la variable *y_pred_val* se almacena la secuencia de estados ocultos, determinado para cada observación de *X_val*, infiriendo por el modelo la categoría a la que pertenece cada sentimiento.

6. EVALUACIÓN DE LA PRECISIÓN Y EFICACIA DEL SISTEMA DESARROLLADO

La realización de las pruebas, se ejecutaron de manera iterativa con una validación cruzada, las cuales fueron empleadas para la valoración del modelo oculto de Markov, donde se valoraron los modelos propuestos con las métricas Accuracy y F1-score, con diferentes números de estados ocultos (n_components), con la finalidad de evaluar el rendimiento de los diferentes modelos ejecutados.

En referencia a la evaluación del modelo, el conjunto de datos se dividió en dos partes, Training para el entrenamiento del modelo y Test para evaluar el rendimiento del modelo [27], con la siguiente distribución:

- Training: 90% de los datos entrenamiento, según la tabla 8, matrices para el conjunto de entrenamiento
- Test: 10% de los datos de validación descritos en la sección 5.4.1, mapeo de sentimientos.

Por consiguiente, en la siguiente tabla, se presentan los resultados destacados de las validaciones realizadas con las métricas Accuracy y F1-score.

Estados Ocultos	Vectorización	Clasificación POS Tagging	Normalización de Datos	Tipo de Covarianza	Accuracy	F1-Score
41	**TF-IDF**	**Si**	**No**	**spherical**	**0.9091**	**0.9051**
30	Bag of Words	Si	No	spherical	0.7222	0.7312
36	TF-IDF	Si	Si	spherical	0.7308	0.7267

Tabla 8: Resultados destacados con las métricas Accuracy y F1-score del dataset IMDb

En los resultados de la evaluación se destacaron tres modelos, uno con excelente rendimiento, dado que su resultado está por encima del valor de 0.85 y dos con valores buenos o aceptables por estar entre el rango de 0.70 y 0.79, otra acotación sobre los resultados es que la relación entre los valores de Accuracy y F1-score, demuestra que el modelo está clasificando bien las clases, incluso si están desbalanceadas.

Por otra parte, se identifica que la clasificación de palabras con POS Tagging (etiquetado de parte de discurso), y el tipo de covarianza esférica (spherical), están presente en los tres modelos destacados.

Asimismo, en la siguiente tabla se presentan los resultados del dataset Yelp, similares a los resultados del dataset anterior.

Estados Ocultos	Vectorización	Clasificación POS Tagging	Normalización de Datos	Tipo de Covarianza	Accuracy	F1-Score
32	**BERT-base**	**Si**	**Si**	**spherical**	**0.7222**	**0.7894**
51	**Bow**	**Si**	**No**	**spherical**	**0.7058**	**0.7028**
47	BERT-base	No	Si	spherical	0.5666	0.8562

Tabla 9: Resultados destacados con las métricas Accuracy y F1-score del dataset Yelp

En donde, los dos primeros resultados destacados son buenos o aceptables, ya que se encuentra entre el rango de 0.70 y 0.79, en el caso del tercer resultado se observa que la métrica Accuracy es baja y la métrica F1-score es alta, lo cual evidencia un desequilibrio importante en el desempeño del modelo, debido a que a pesar de que tiene un buen balance entre precisión y recall, pero bajo en el acierto total.

Finalmente, semejante que el dataset anterior, el tipo de covarianza esférica (spherical) está presente en los tres modelos y la clasificación de palabras con POS Tagging (etiquetado de parte de discurso), está sólo presente en los dos modelos destacados.

7. CONCLUSIONES

- El proceso de limpieza, normalización y lematización del texto fue esencial para garantizar la calidad de los datos utilizados, del mismo modo, las técnicas aplicadas, como la eliminación de stopwords y la tokenización, redujeron el ruido del texto y permitieron una mejor representación semántica de las reseñas. Asimismo, las vectorizaciones o los métodos de representación textual, Bag of Words, TF-IDF y Word Embeddings contribuyeron en la capacidad del modelado para capturar patrones relevantes del lenguaje, y como medida de experimentación se adicionaron cuatro vectorizaciones adicionales, GloVe 300d, Word2Vec-384d, FastText-300 y BERT-base, con el propósito de validar y optimizar el rendimiento de los modelos probabilísticos aplicados.
- Los resultados obtenidos de las técnicas de normalización de textos y el uso de etiquetado gramatical *POS Tagging,* proporcionaron estabilidad y significado a los datos de entrada, lo que permitió en el modelo capturar mejor las dependencias secuenciales entre palabras, lo que resultó determinante para mejorar la convergencia del algoritmo de entrenamiento y la calidad de los resultados.
- En el entrenamiento y la validación de los Modelos Ocultos de Markov, se evidenció que las covarianzas *full* y *tied* exigieron un alto costo computacional debido a la complejidad de las operaciones matriciales, al contrario, las covarianzas *spherical* y *diag* ofrecieron un mejor equilibrio entre rendimiento y eficiencia, reduciendo significativamente los tiempos de procesamiento sin deteriorar la precisión del modelo.
- Los resultados de desempeño de las métricas de evaluación en el rendimiento del modelo F1-Score y Accuracy, reflejaron un equilibrio entre precisión y exhaustividad, en el caso de F1-Score fue aproximado a 0.75 para los dos dataset, y en el caso de Accuracy fue superior a 0.85 para el dataset IMDb, lo cual demuestra que los modelos HMM proporcionan un desempeño competitivo para el análisis de sentimientos en reseñas.
- La integración de los Modelos Ocultos de Markov con técnicas de Procesamiento del Lenguaje Natural, representa un sistema efectivo para la clasificación automática de sentimientos. Además, la combinación de características lingüísticas y secuenciales permitió alcanzar resultados robustos y consistentes, validando la aplicabilidad de los HMM en la clasificación de análisis textual.
- Finalmente, se elegio los Modelos Ocultos de Markov (HMM), debido a que permiten modelar de forma natural la secuencia y evolución temporal del sentimiento dentro de un texto, a diferencia de modelos tradicionales de clasificación que analizan cada reseña de forma aislada, los HMM capturan estructuras latentes y transiciones entre estados emocionales, lo cual es especialmente relevante en reseñas donde el sentimiento puede cambiar a lo largo de las oraciones. Además, los HMM se adaptan bien a escenarios donde los estados reales (sentimientos) no son observables directamente, pero sí pueden inferirse a partir de observaciones visibles (palabras, vectores de texto), lo que encaja de manera directa con el problema del análisis de sentimientos.

REFERENCIAS BIBLIOGRÁFICAS

[1] "La reseña", Escuela de filosofía y humanidades, Departamento de lectura y escritura académica, Universidad Sergio Arboleda, 2014.

[2] 40 Online Review Statistics 2025 (Latest Data)," *DemandSage*. [Online]. Disponible en: https://www.demandsage.com/online-review-statistics/

[3] V. Odumuyiwa and U. Osisiogu, "A Systematic Review on Hidden Markov Models for Sentiment Analysis," 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2019, pp. 1-7.

[4] CERN, "The birth of the web," CERN, [Online]. Disponible en: https://home.cern/science/computing/birth-web

[5] La Voz (2023, Feb 6). El dato actualizado: cuántas páginas web hay en toda la internet. [Online]. Disponible en: https://www.lavoz.com.ar/tecnologia/el-dato-actualizado- cuantas-paginas-web-hay-en-toda-la-web/#:~:text=Esta%20evoluci%C3%B3n%20de%20Internet%2C%20analizada,%C3%BAnicos%20y%2012.156.700%20servidores.

[6] K. P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning," Foundations and Trends in Machine Learning, vol. 1, no. 1–2, pp. 1–305, 2006. [Online]. Disponible en: https://doi.org/10.1561/1500000001

[7] I. Hatzilygeroudis, M Paraskevas, S Kardakis, I Perikos, "Hidden Markov Models for Sentiment Analysis in Social Media", IEEE Xplore, pp. 130-135, May 2019.

[8] A. Merline Lawrence y A. Pradhan Adhikari, "Sentiment Analysis: Methods and Application Using Machine Learning in Different Fields", IEEE, 2019.

[9] G. Jabbour, L. Maldonado y M. Sarmiento, "Un sistema híbrido basado en modelos ocultos de markov y máquinas de vectores de soporte con aprendizaje hacia adelante para reconocimiento de fonos en habla continua venezolana", *Ingeniería UC*, vol. 18, pp. 7-16, Dic 2011.

[10] D. Macas y W. Padilla, "*Estudio de los modelos ocultos de markov y desarrollo de un prototipo para el reconocimiento automático del habla*", Proyecto de grado, Facultad de ingeniería, Universidad Politécnica Salesiana, Cuenca, Ecuador, 2012.

[11] Y. Khalifa, D. Mandic y E. Sejdić, "*A review of hidden markov models and recurrent neural networks for event detection and localization in biomedical signals*", *ScienceDirect*, vol. 69, pp. 52-72, May 2021.

[12] A. Hurtado (2017, Jun 17). ¿Qué son y cómo funcionan los portales Web?. [Online]. Disponible en: https://info.netcommerce.mx/funcionan-los-portales-web/

[13] F. Rubio (2019, Jun 2). ¿Cómo son tus reseñas en Internet?. [Online]. Disponible en: https://www.expacioweb.com/como-son-tus-resenas-en-internet/

[14] A. Jain, "Natural Language Processing for Beginners," *Data Science Salon*, May 2022. [Online]. Disponible en: https://roundtable.datascience.salon/natural-language-processing-for-beginners

[15] S. Bird, E. Klein, y E. Loper, *Natural Language Processing with Python*. Gravenstein: O'Reilly Media, Inc, 2009.

[16] GAMCO, "Minería de textos: concepto y definición," *GAMCO Inteligencia Artificial*, [En línea]. Disponible en: https://gamco.es/glosario/mineria-de-textos/.

[17] A. Fernández, *Introducción a la minería de texto y análisis de sentimiento con R*: Limencop S.L. 2023.

[18] A. González (2007, Apr 11). Extracción y recuperación de la información II: Clasificación supervisada. [Online]. Disponible

en:
https://web.archive.org/web/20110719165554/http://supervisadaextraccionrecuperacioninformacion.iespana.es/
[19] J. Martin, D. Jurafsky, Speech and Language Processing. Stanford: Prentice Hall, 2023.
[20] J. Bulla, "Application of Hidden Markov Models and Hidden Semi-Markov Models to Financial Time Series", Ph.D. dissertation, Georg-August-University of Göttingen, Göttingen. 2006.
[21] S. Chandrashekar, "Unraveling the Enigma: Exploring the Hidden Markov Model," *Medium*, 22-mar-2024. [En línea]. Disponible en: https://medium.com/@saivarunchandrashekar/unraveling-the-enigma-exploring-the-hidden-markov-model-7d1e31272ecd.
[22] O. Ramos, Métricas de Evaluación. Tópicos de Ingeniera Mecatrónica. Perú: Universidad de Ingeniera y Tecnóloga- UTEC, 2020.
[23] C. Carvajal, "ACTIVISM: Plataforma para el análisis de información sobre de flujos de texto en redes sociales", trabajo de maestría, Universidad de los Andes, Bogotá D.C. 2019.
[24] J. Bobadilla, Machine Learning y Deep Learning. Usando Python, Scikit y Keras, 1a. Madrid: RAMA, 2020.
[25] "sklearn.preprocessing.StandardScaler — documentación de scikit-learn - 0.24.2," Scikit-learn en Español (qu4nt.github.io). [Online]. Disponible en: https://qu4nt.github.io/sklearn-doc-es/modules/generated/sklearn.preprocessing.StandardScaler.htm
[26] B. Judah, "Removing stop words with NLTK library in Python," Analytics Vidhya, Oct. 20, 2021. [Online]. Disponible en: https://medium.com/analytics-vidhya/removing-stop-words-with-nltk-library-in-python-f33f53556cc1.
[27] IBM Corporation, "Conceptos básicos sobre modelado - Documentación de IBM." https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=modeling-overview.
[28] E. Clarke, "Etiquetado de POS con NLTK y fragmentación en NLP [EJEMPLOS]," Guru99, 3 de octubre de 2024. [En línea]. Disponible en: https://www.guru99.com/es/pos-tagging-chunking-nltk.html
[29] Técnicas de extracción de características en texto: BoW y TFIDF, Toolify, [Online]. Disponible en: https://www.toolify.ai/es/ai-news-es/tcnicas-de-extraccin-de-caractersticas-en-texto-bow-y-tfidf-2283268.
[30] Técnicas clave para procesamiento de texto NLP," OpenWebinars, [Online]. Disponible en: https://openwebinars.net/blog/tecnicas-clave-para-procesamiento-texto-nlp/.
[31] *hmmlearn contributors, hmmlearn: Hidden Markov Models in Python, [Online]. Disponible en: https://hmmlearn.readthedocs.io/en/latest/index.html.*
[32] I. Kononenko and M. Kukar, "Chapter 12 - Cluster Analysis," in Machine Learning and Data Mining, I. Kononenko and M. Kukar, Eds., Cambridge, UK: Woodhead Publishing, 2007, pp. 321–358. ISBN: 9781904275213.
[33] Pierian Training, "Viterbi Algorithm Implementation in Python: A Practical Guide," Pierian Training, [Online]. Disponible en: https://pieriantraining.com/viterbi-algorithm-implementation-in-python-a-practical-guide/.
[34] J. Shaik, "Choosing the Best Model: A Friendly Guide to AIC and BIC," Medium, Sep. 19, 2023. [Online]. Disponible en: https://medium.com/@jshaik2452/choosing-the-best-model-a-friendly-guide-to-aic-and-bic-af220b33255f

8. ANEXOS

Anexo N.1: Gráficos de líneas de AIC y BIC del dataset IMDb

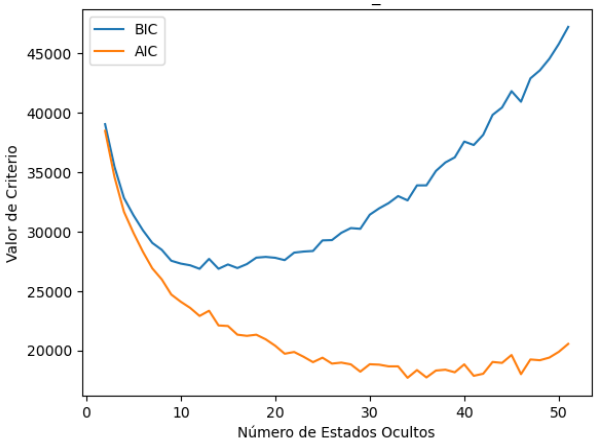

Figura N. 12: Bow, spherical, y POS Tagging

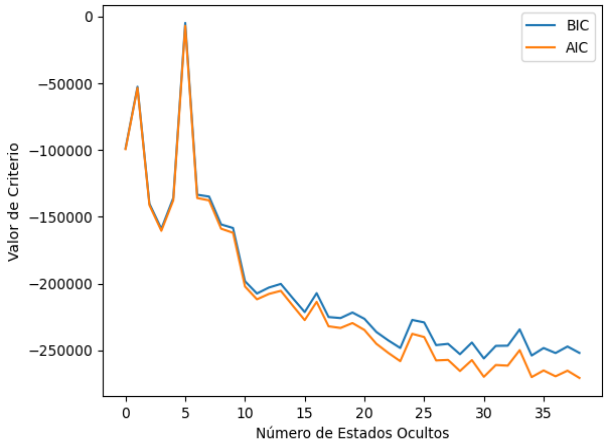

Figura N. 6: BoW, full, normalizado y POS Tagging

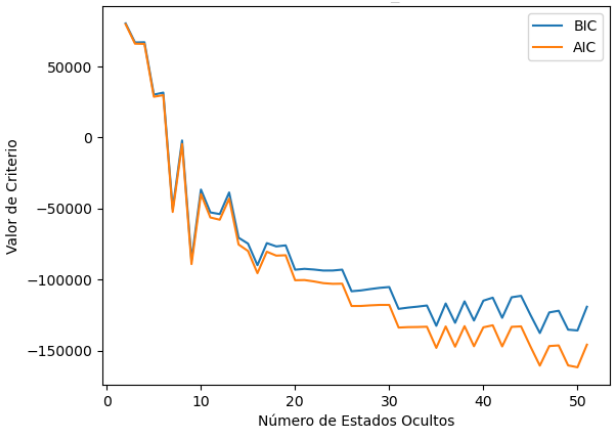

Figura N. 7: Bow, tied, normalizado y POS Tagging

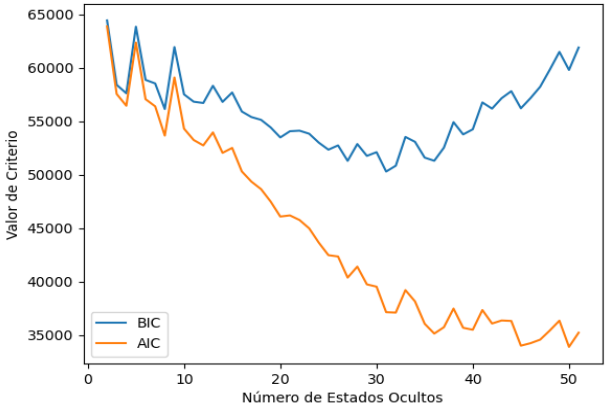

Figura N. 8: Bow, spherical, normalizado y POS Tagging

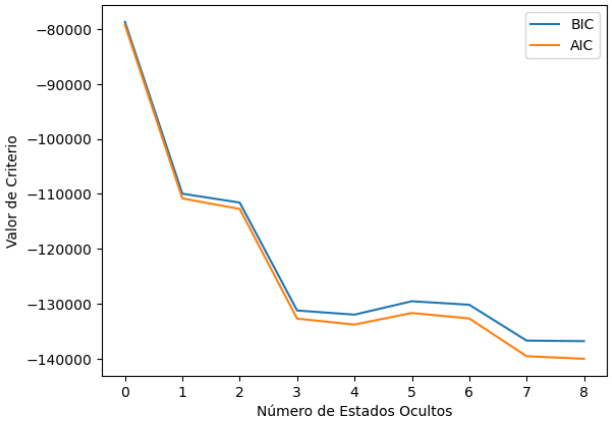

Figura N. 9: Bow, diag y POS Tagging

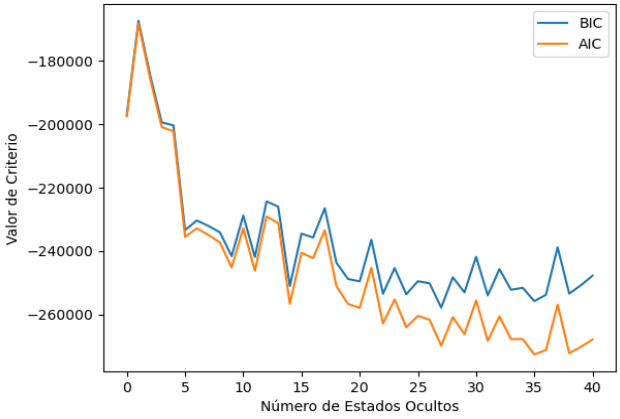

Figura N. 10: Bow, full, y POS Tagging

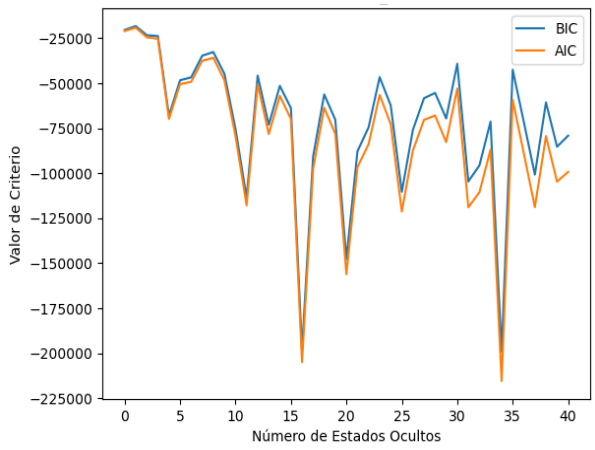

Figura N. 11: Bow, tied, y POS Tagging

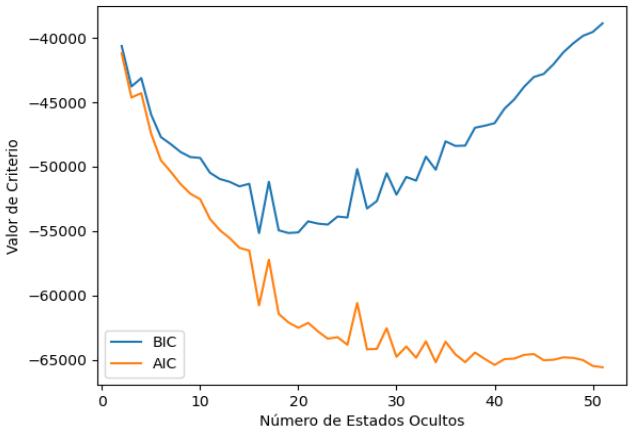

Figura N. 20: TF-IDF, spherical y POS Tagging

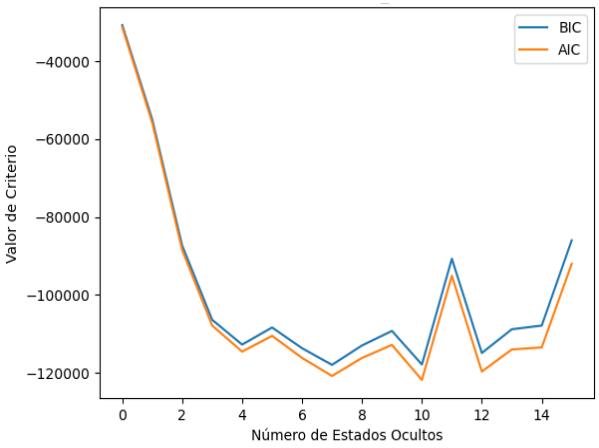

Figura N. 13: TF-IDF, diag, normalizado y POS Tagging

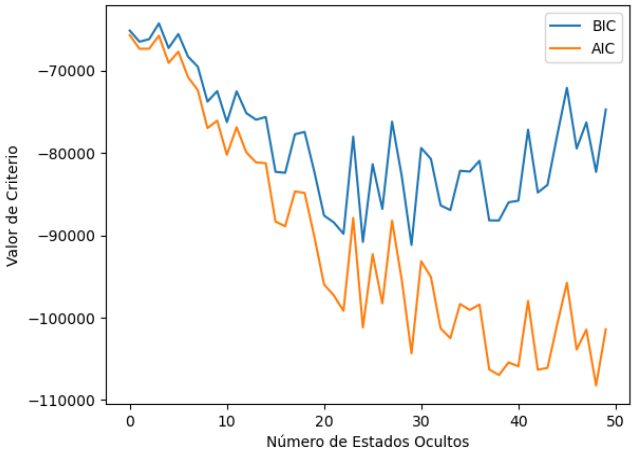

Figura N. 19: TF-IDF, tied y POS Tagging

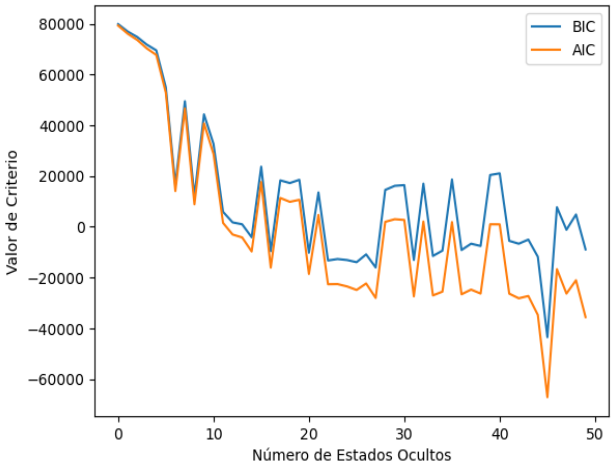

Figura N. 15: TF-IDF, tied, normalizado y POS Tagging

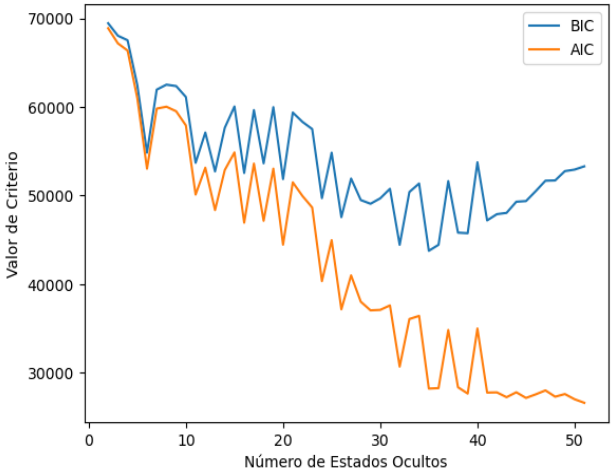

Figura N. 16: TF-IDF, spherical, normalizado y POS Tagging

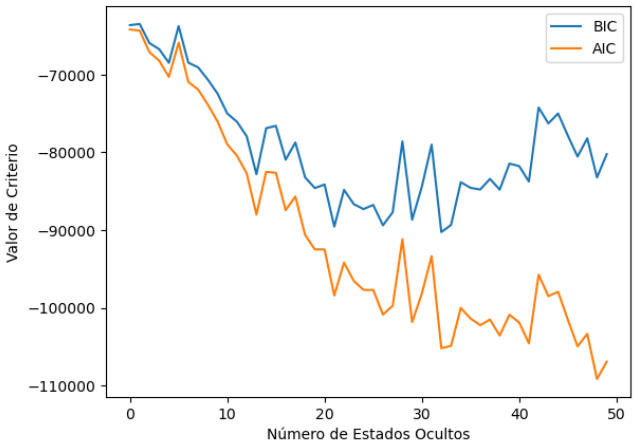

Figura N. 17: TF-IDF, diag y POS Tagging

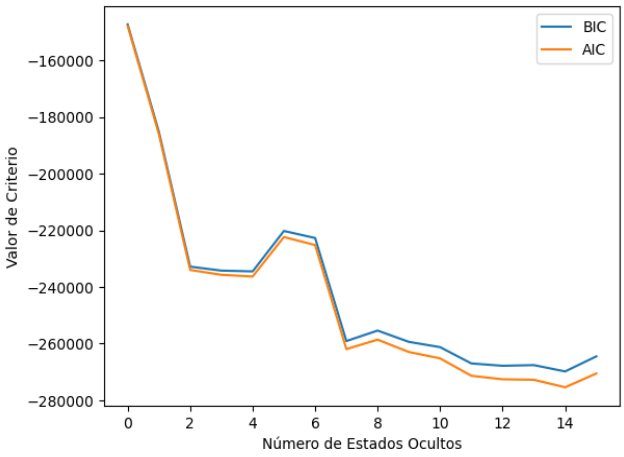

Figura N. 18: TF-IDF, full y POS Tagging

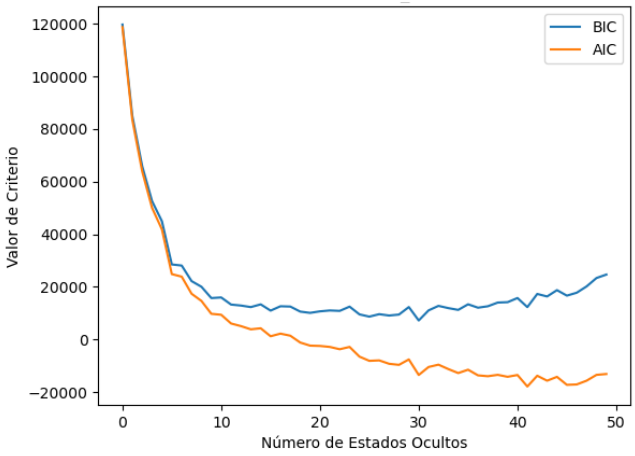

Figura N. 21: Embedding, diag, normalizado y POS Tagging

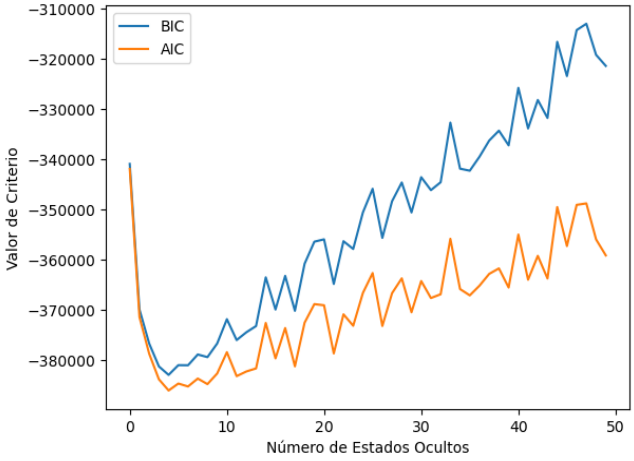

Figura N. 28: Embedding, spherical y POS Tagging

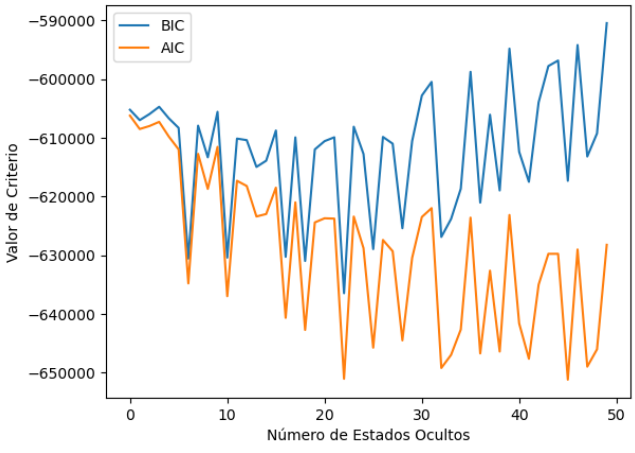

Figura N. 23: Embedding, tied, normalizado y POS Tagging

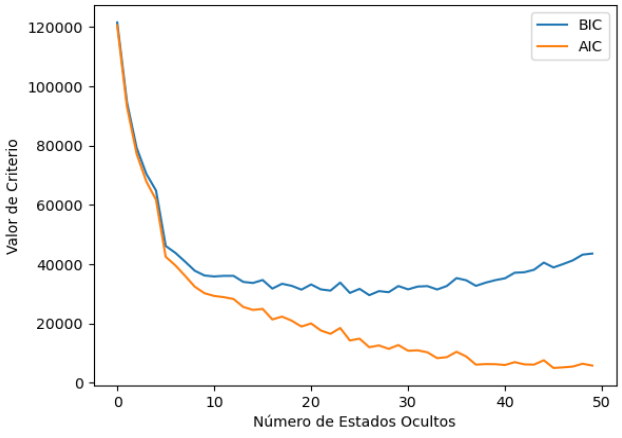

Figura N. 24: Embedding, spherical, normalizado y POS Tagging

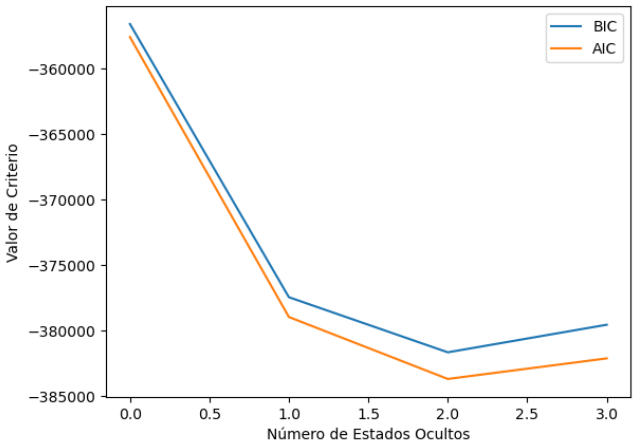

Figura N. 25: Embedding, diag y POS Tagging

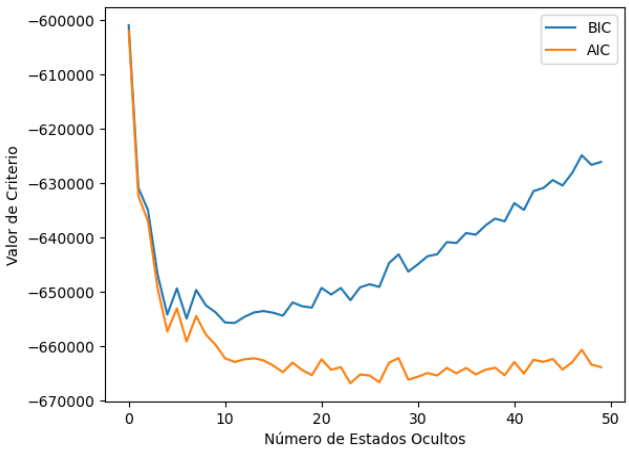

Figura N. 26: Embedding, diag y POS Tagging

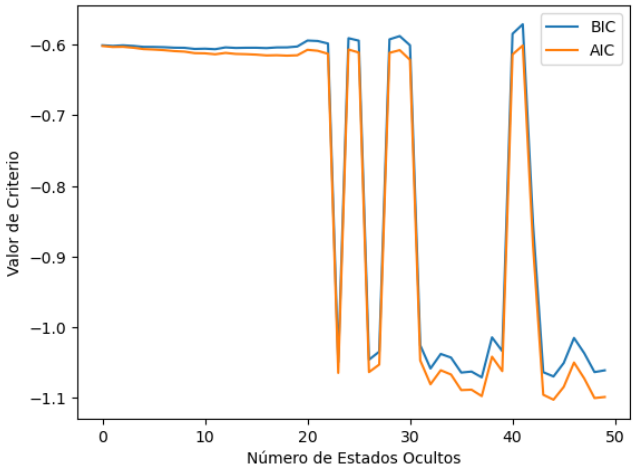

Figura N. 27: Embedding, tied y POS Tagging

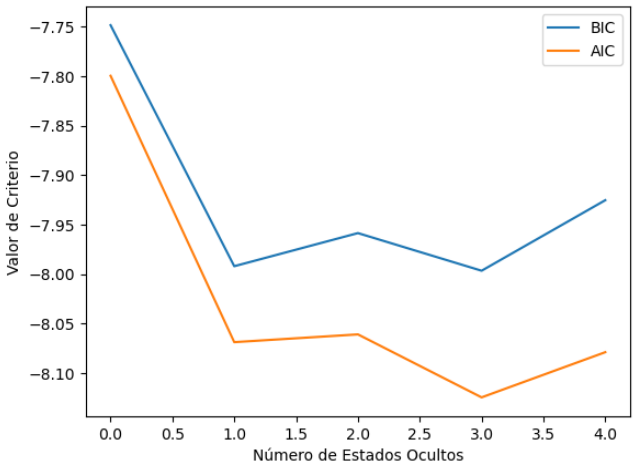

Figura N. 34: Bow y spherical

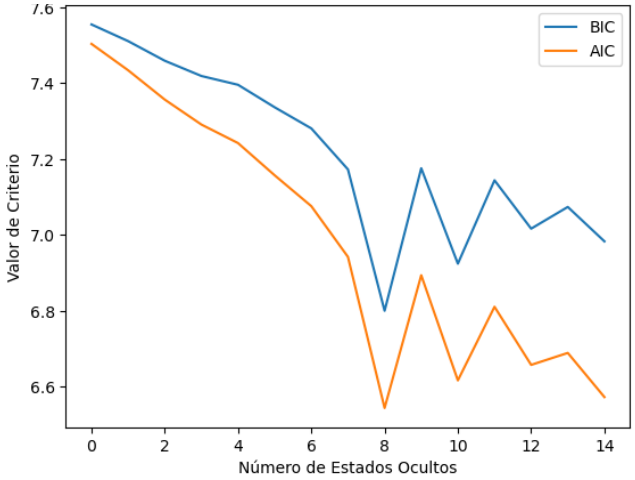

Figura N. 29: BoW, diag y normalizado

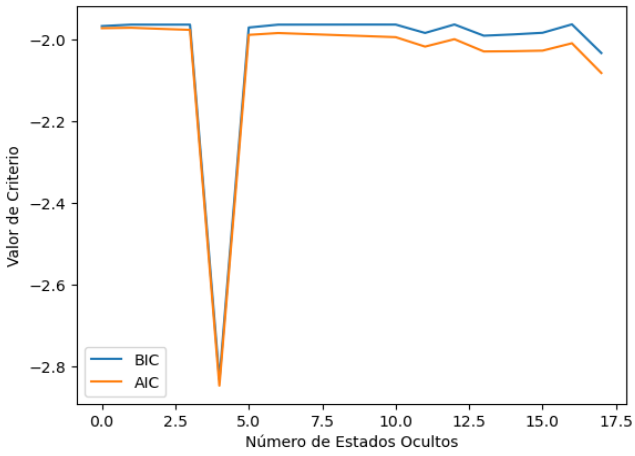

Figura N. 30: BoW, full y normalizado

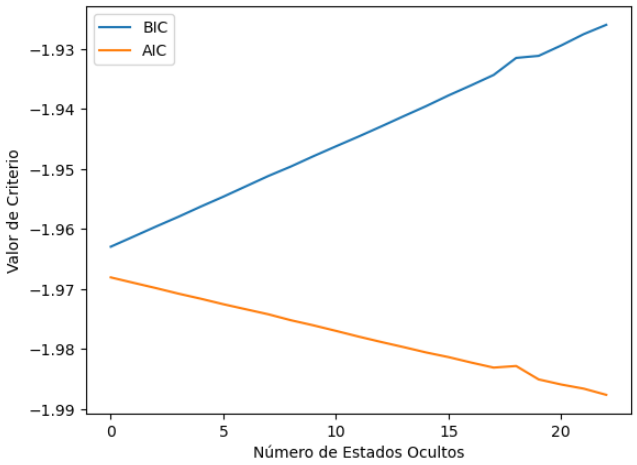

Figura N. 31: BoW, tied y normalizado

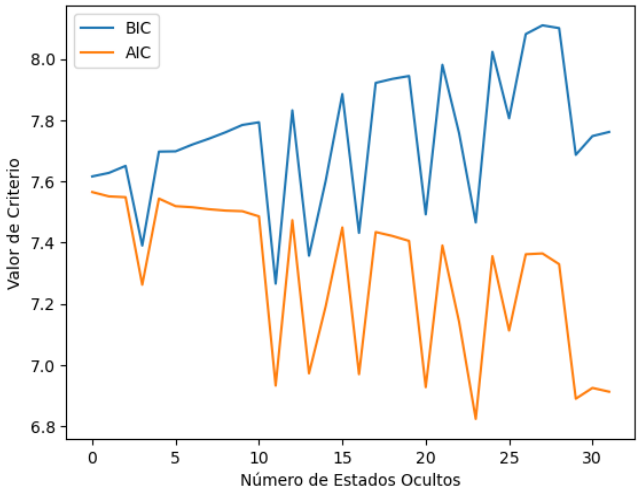

Figura N. 32: BoW, spherical y normalizado

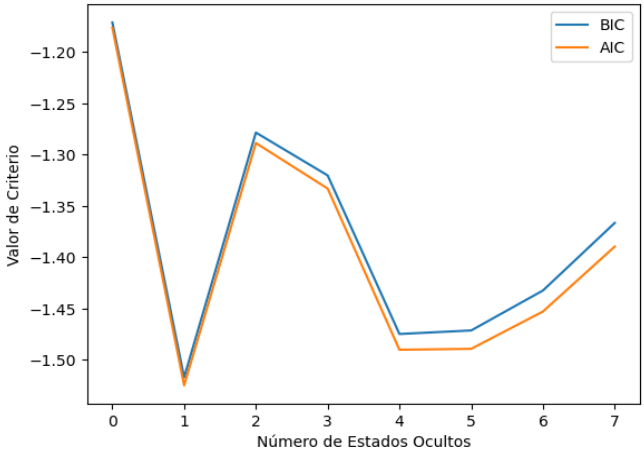

Figura N. 33: Bow y Diag

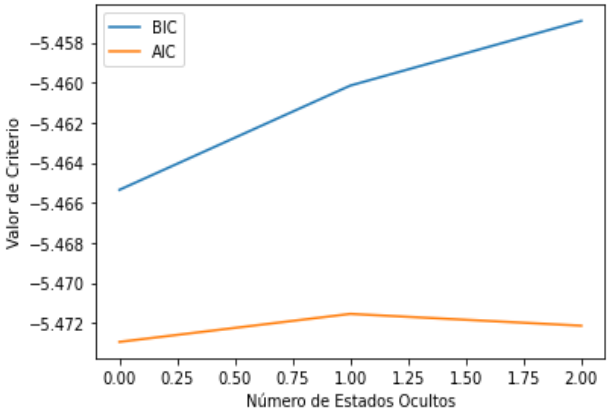

Figura N. 50: Word2Vec-384d, tied y POS Tagging

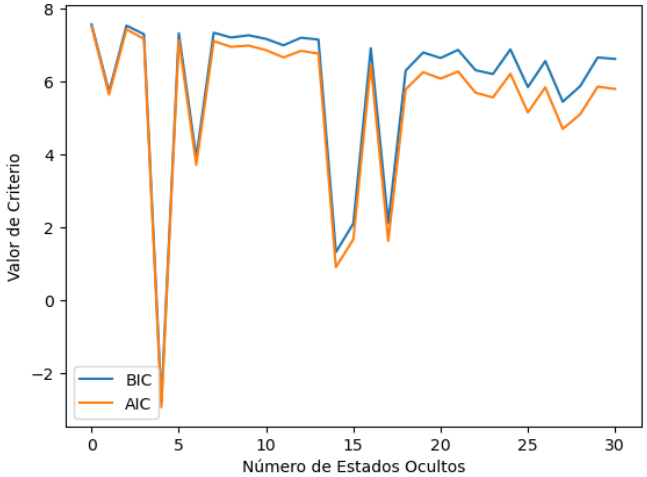

Figura N. 35: TF-IDF, diag y normalizado

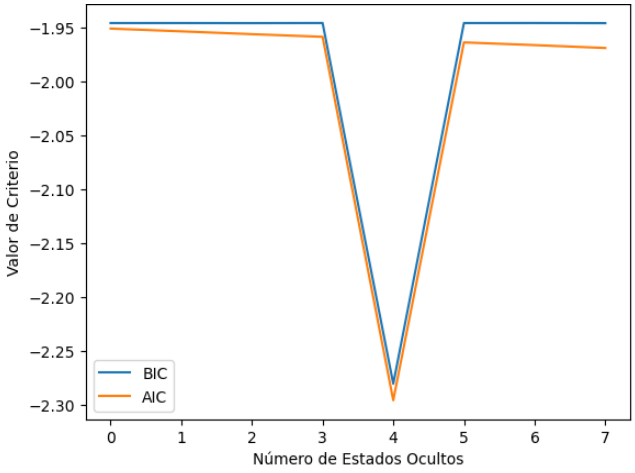

Figura N. 36: TF-IDF, full y normalizado

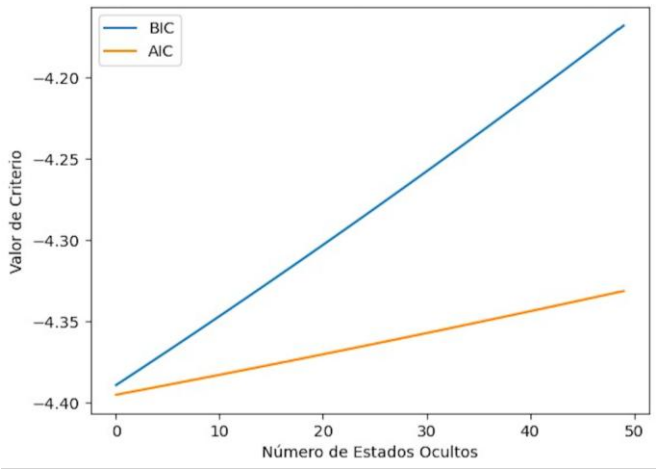

Figura N. 37: Glove-300d, tied y POS Tagging

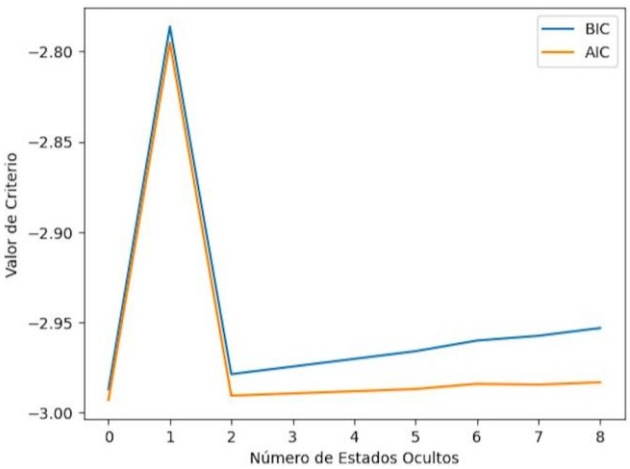

Figura N. 38: Glove-300d, spherical y POS Taggin

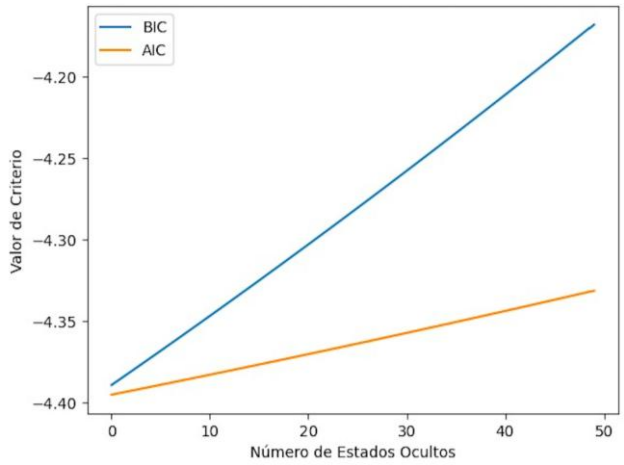

Figura N. 39: Glove-300d, diag y POS Tagging -Esc

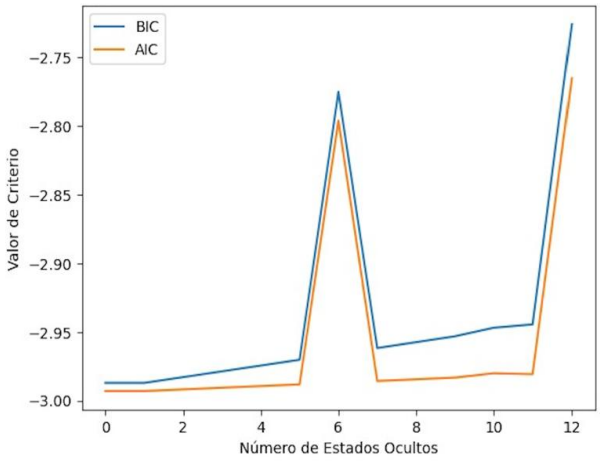

Figura N. 40: Glove-300d, spher y POS Tagging- Esc

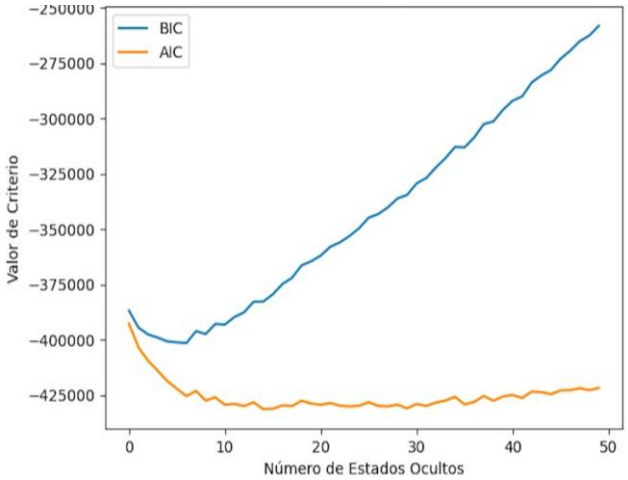

Figura N. 41: Glove-300d, Sph y POS Tagging -Esc

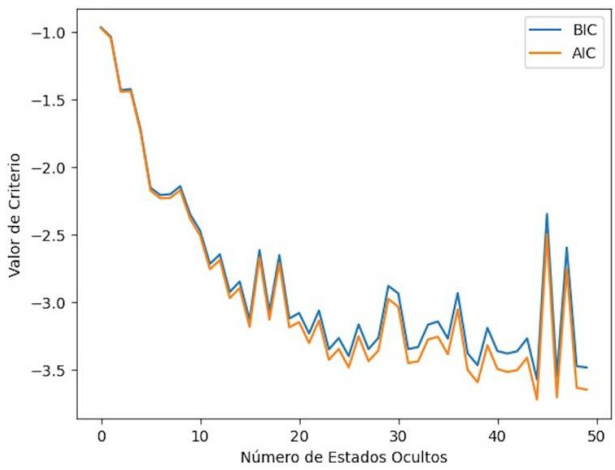

Figura N. 42: Glove-300d, full y POS Tagging -Esc

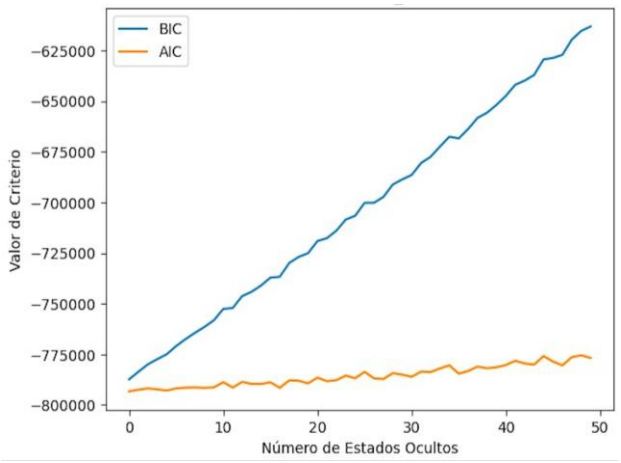

Figura N. 43: Glove-300d, tied y Lematizado_Plano

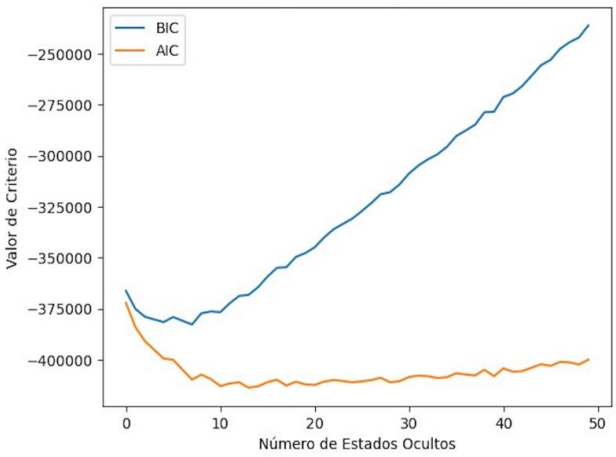

Figura N. 44: Glove-300d, sph y Lemat_Plano

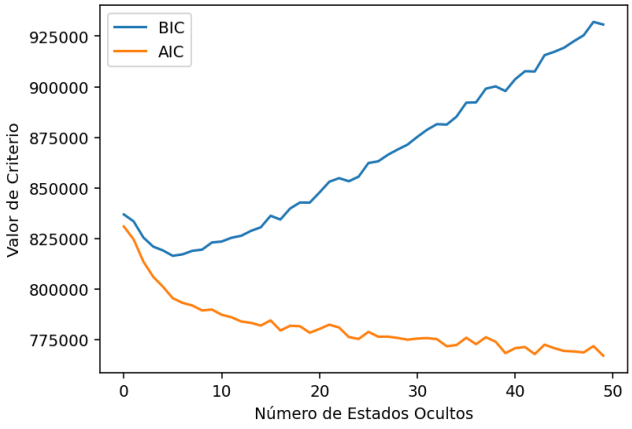

Figura N. 45: Glove-300d, diag y Lematizado_Plano- Esc

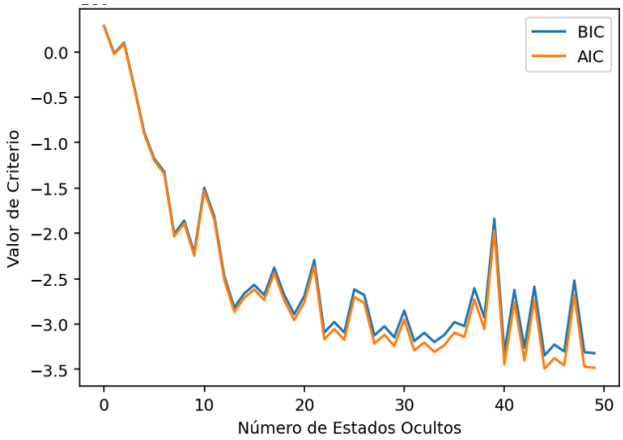

Figura N. 46: Glove-300d, full y Lemat_Plano-Esc

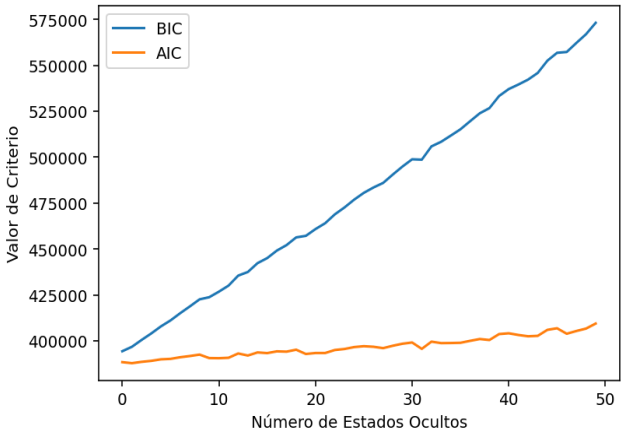

Figura N. 47: Glove-300d, tied y Lemat_Plano- Esc

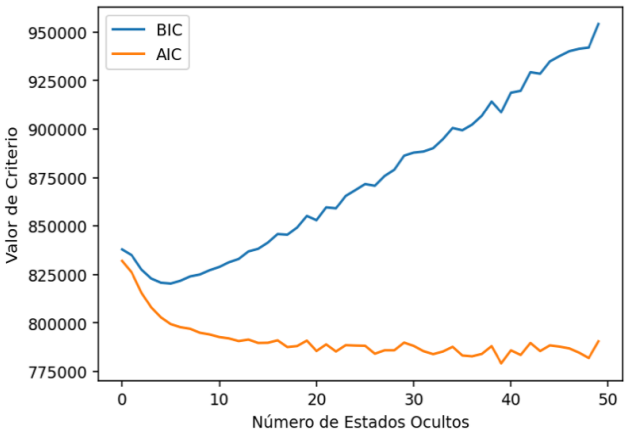

Figura N. 48: Glove-300d, sphl y Lemat_Plano-Esc

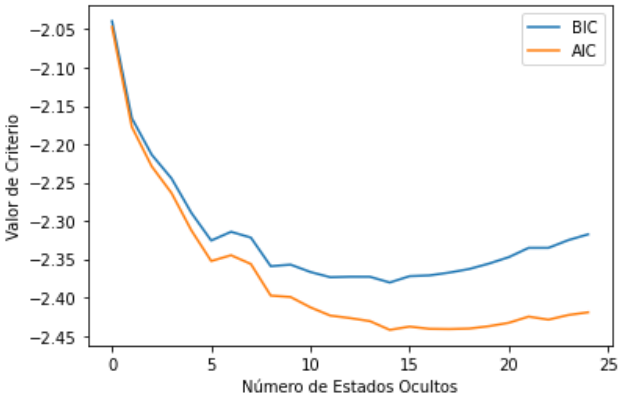

Figura N. 51: Word2Vec-384d, spherical y POS Tag

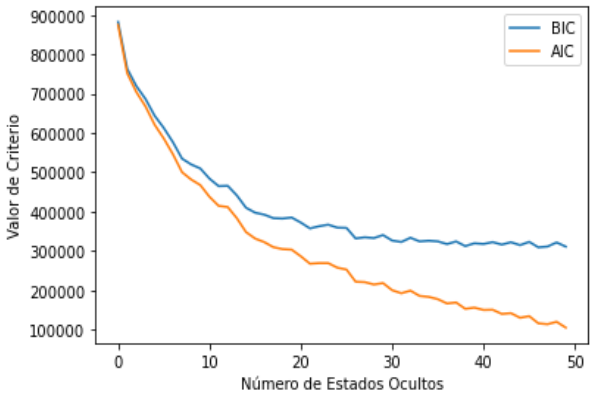

Figura N. 52: Word2Vec-384d, diag y POS Tag- Esca

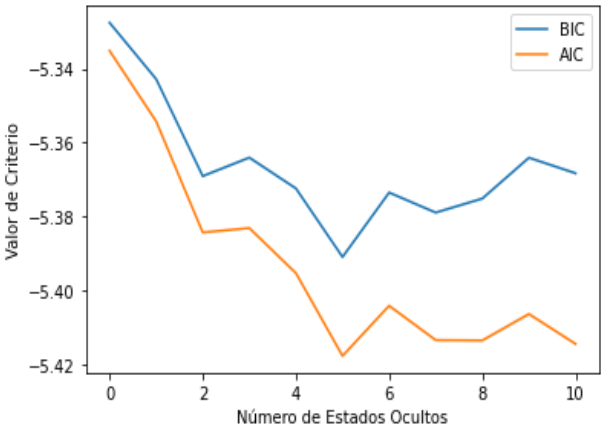

Figura N. 53: Word2Vec-384d, full y POS Tagging-Escal

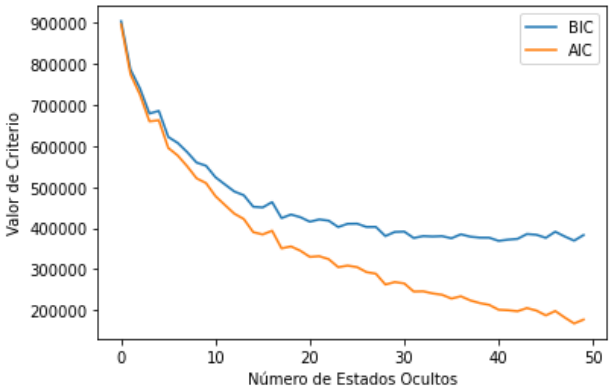

Figura N. 54: Word2Vec-384d, spherical y POS Tagging- Escal

Anexo N.2: Gráficos de líneas de AIC y BIC del dataset Yelp

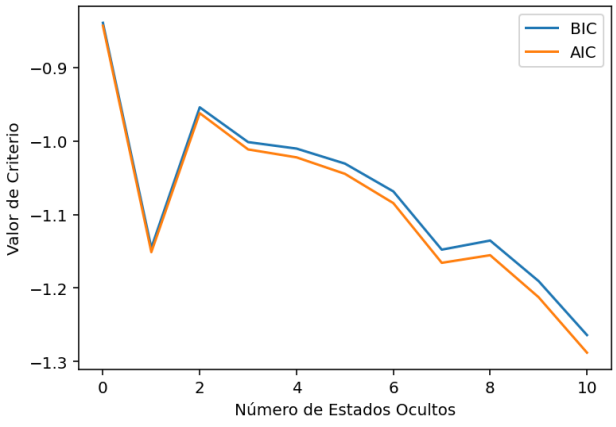

Figura N. 61: BoW, diag y Texto_Lem_Plano - Sin Escal

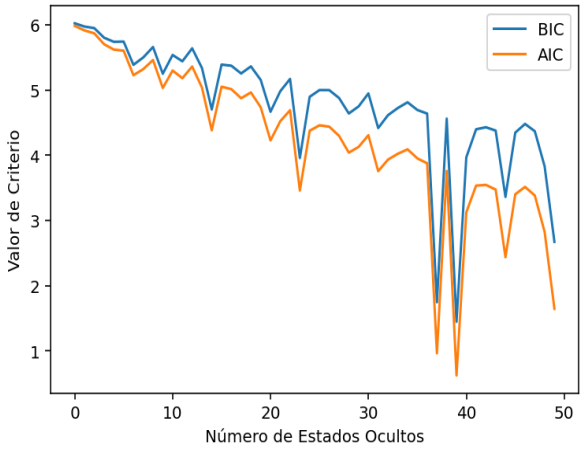

Figura N. 56: BoW, diag y Texto_Lemat_Plano-Esc

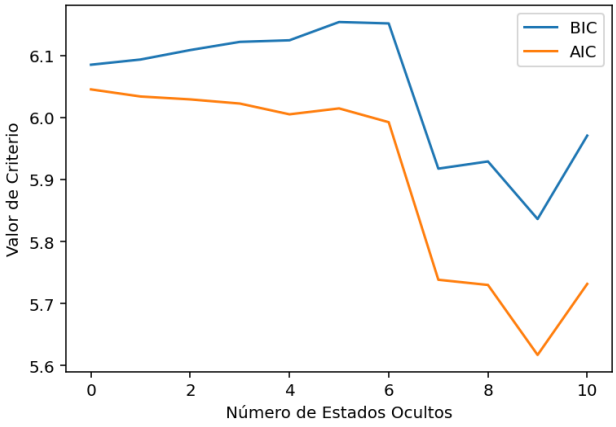

Figura N. 57: BoW, spherical y Texto Lemat Plano-Esc

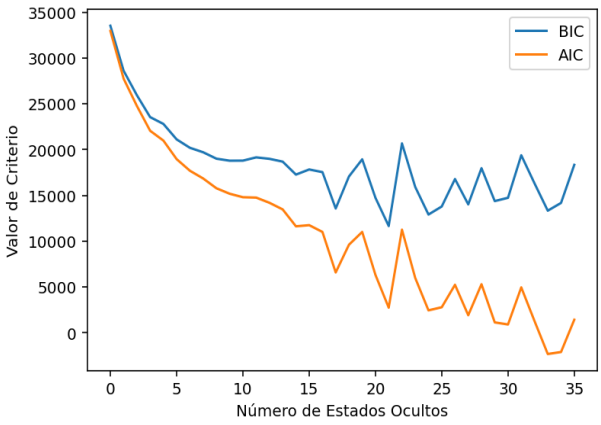

Figura N. 58: BoW, sph y POS Tagging- Sin Escal

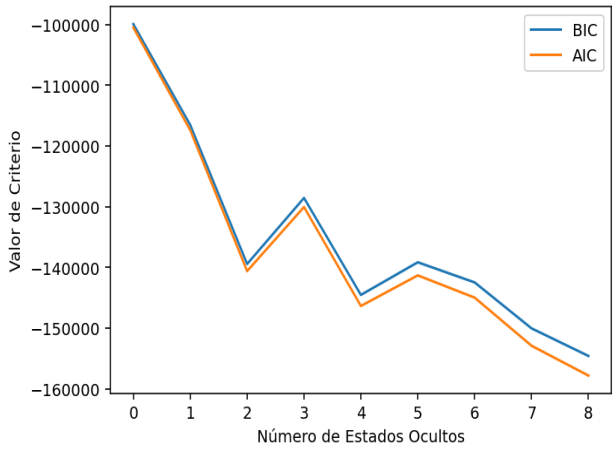

Figura N. 59: BoW, diag y POS Tagging- Sin Escal

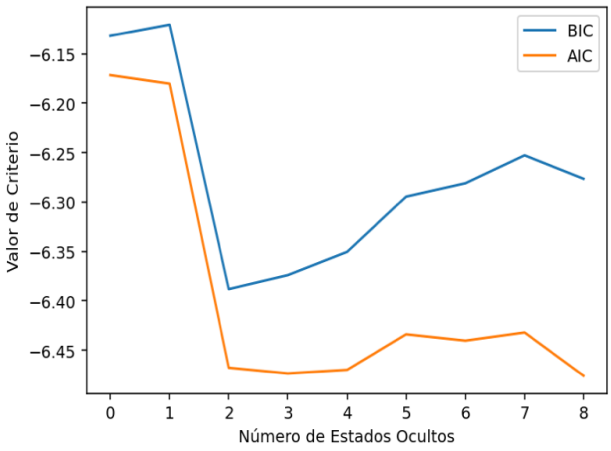

Figura N. 60: BoW, sph y Texto_Lem_Plano - Sin Escal

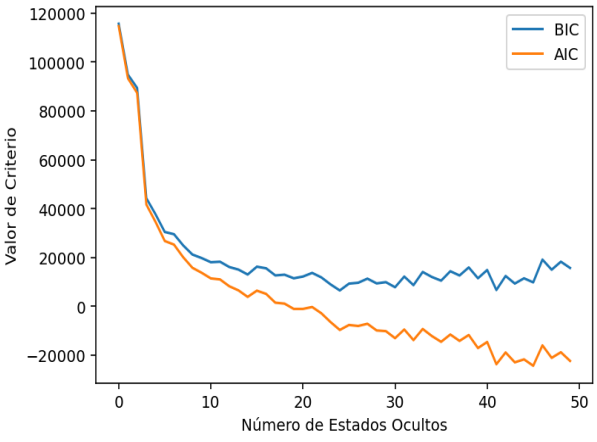

Figura N. 71: Embedding, Sphe y Texto POS -Esc

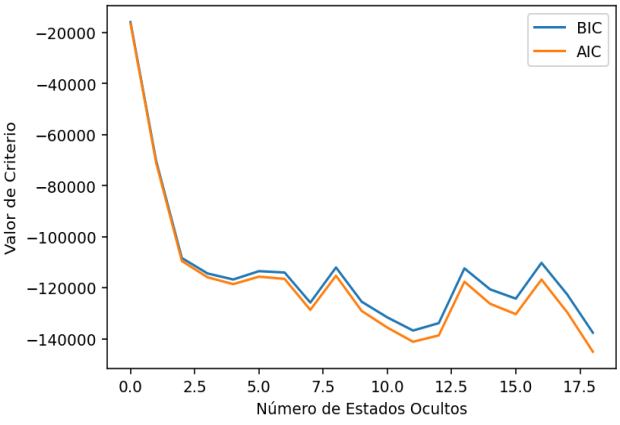

Figura N. 62: TF-IDF, diag y POS Tagging- Esc

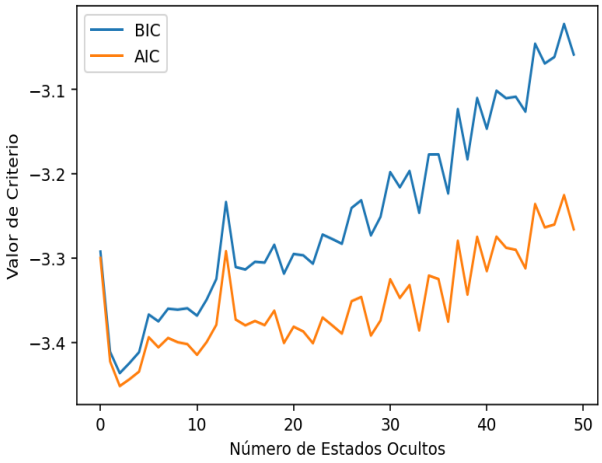

Figura N. 82: Word2Vec-384d, Sph y Texto POS – Sin Esc

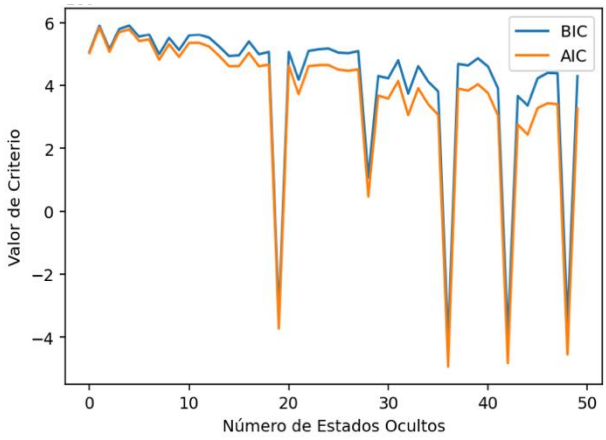

Figura N. 64: TF-IDF, diag y Texto_Lem_Plano

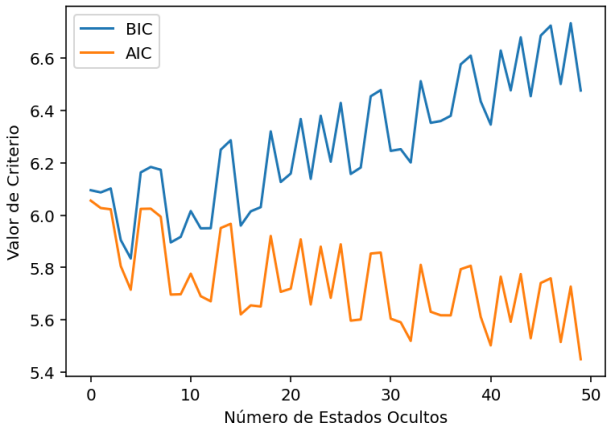

Figura N. 65: TF-IDF, sph y Texto_Lem_Plano-Esc

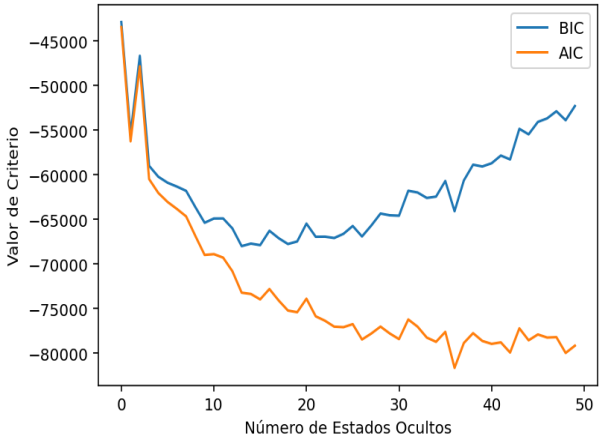

Figura N. 66: TF-IDF, Sph y Texto POS -Sin Esc

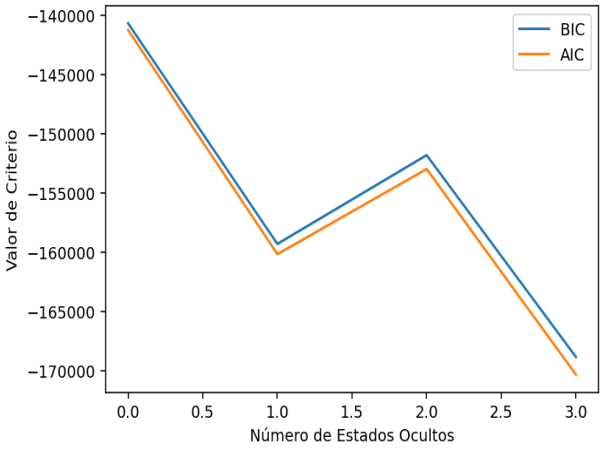

Figura N. 67: TF-IDF, Diag y Texto POS -Sin Esc

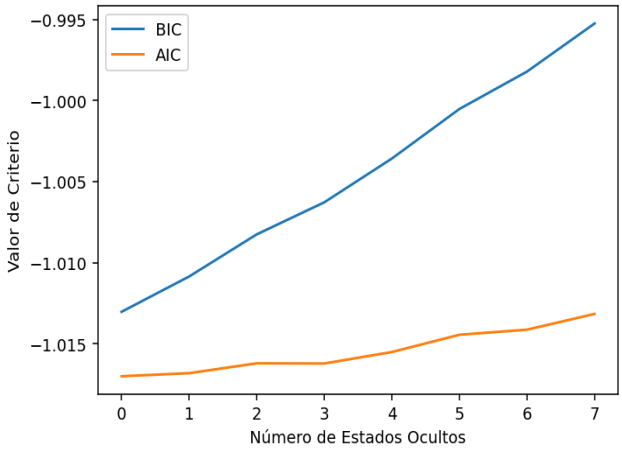

Figura N. 68: TF-IDF, Sph y Texto Lem -Sin Esc

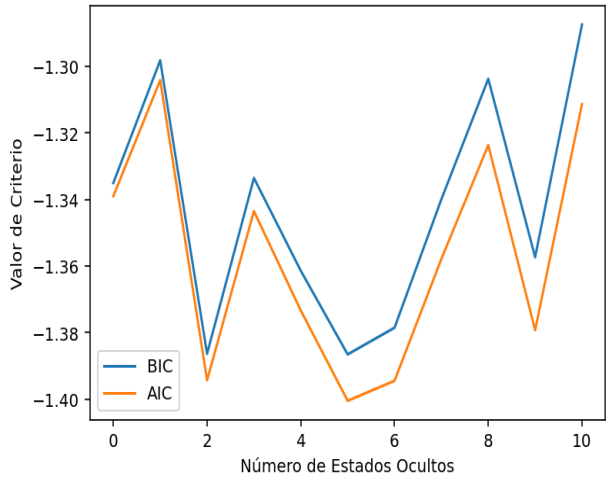

Figura N. 69: TF-IDF, Diag y Texto Lem -Sin Esc

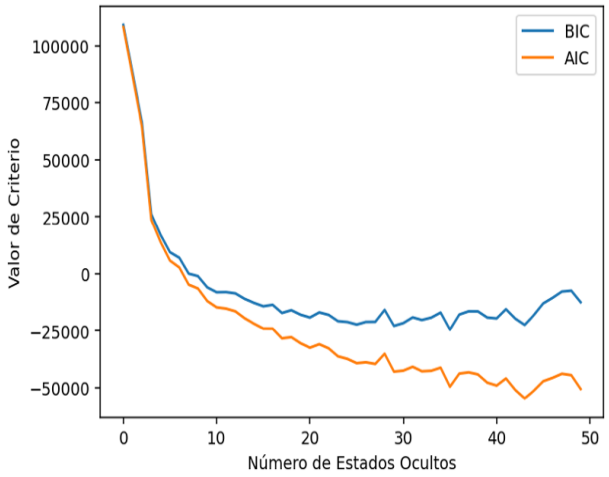

Figura N. 70: Embedding, Diag y Texto POS -Esc

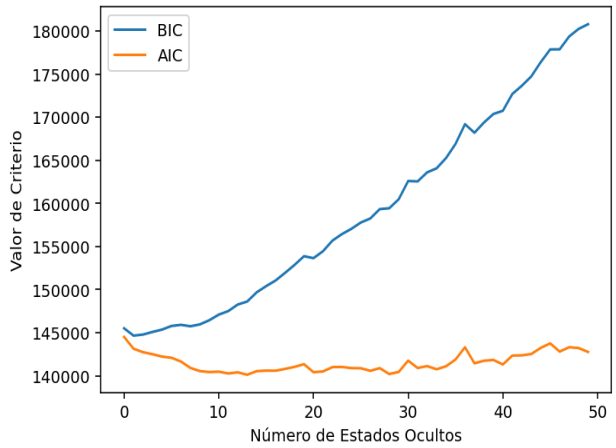

Figura N. 73: Embedding, Sph y Texto Lemat -Esc

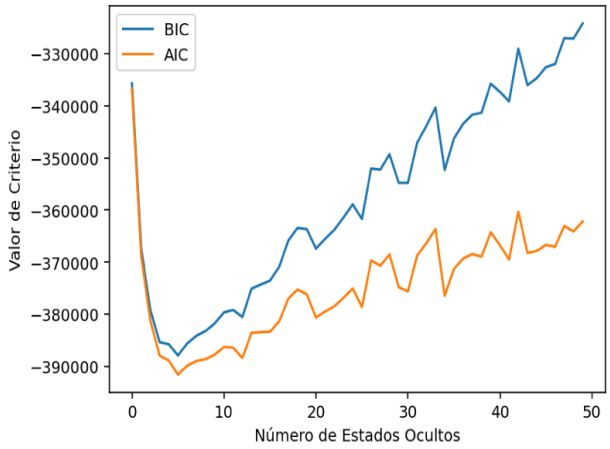

Figura N. 74: Embedding, Sph y Texto POS -Sin Esc

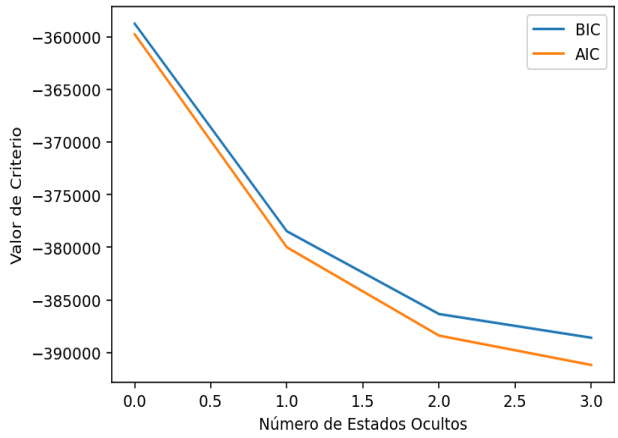

Figura N. 75: Embedding, Diag y Texto POS -Sin Esc

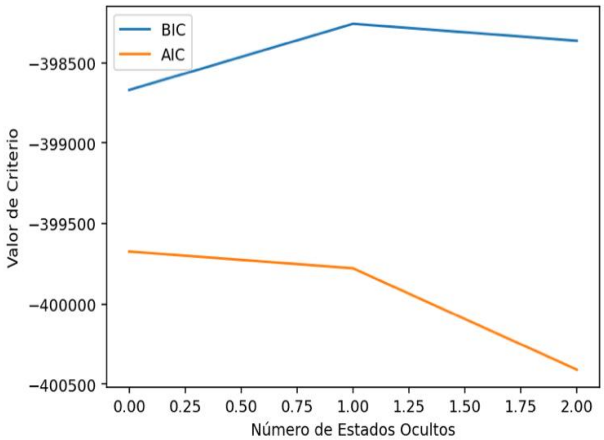

Figura N. 76: Embedding, Sph y Texto Lemat -Sin Esc

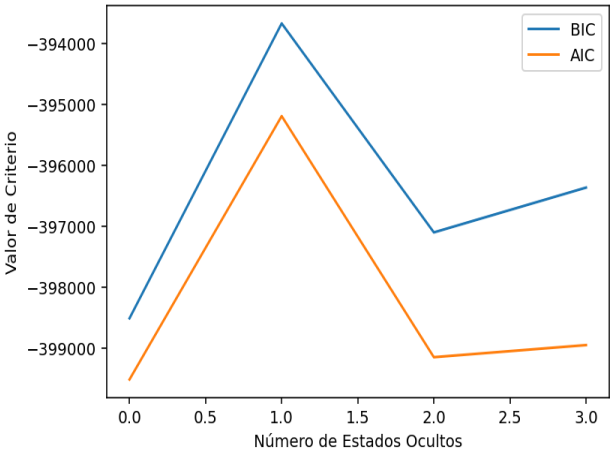

Figura N. 77: Embedding, Diag y Texto Lemat -Sin Esc

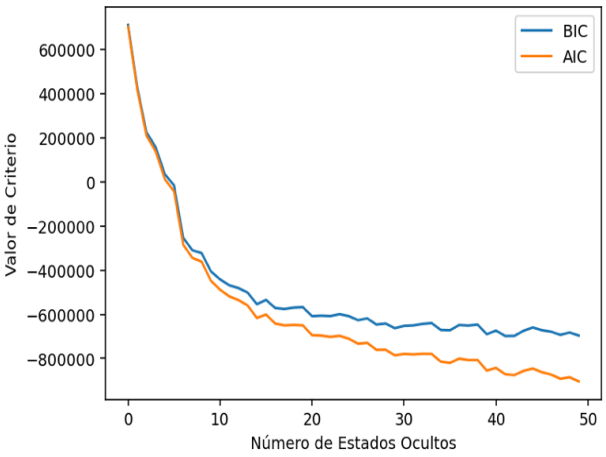

Figura N. 78: Word2Vec-384d, Diag y Texto POS – Esc

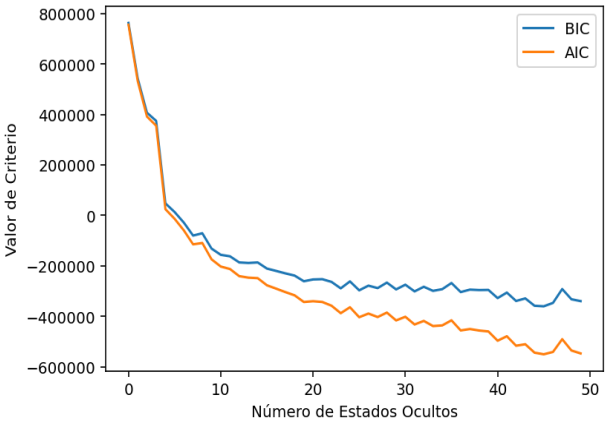

Figura N. 79: Word2Vec-384d, Sph y Texto POS – Esc

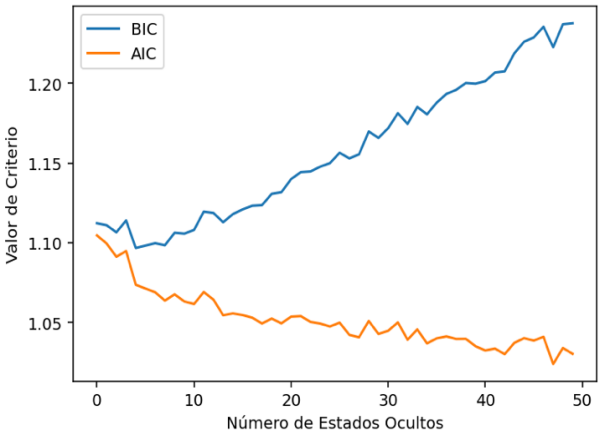

Figura N. 80: Word2Vec-384d, Diag y Texto Lema – Esc

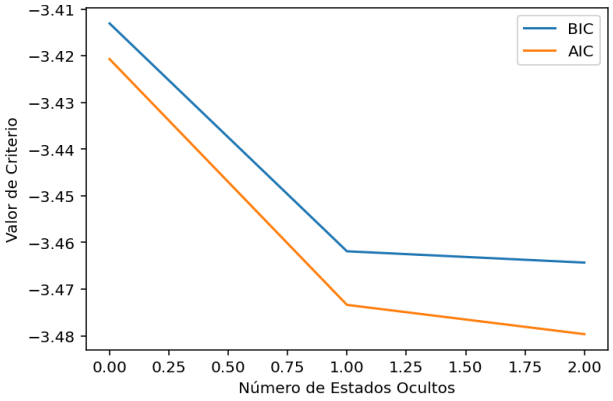

Figura N. 83: Word2Vec-384d, Diag y Texto POS – Sin Esc

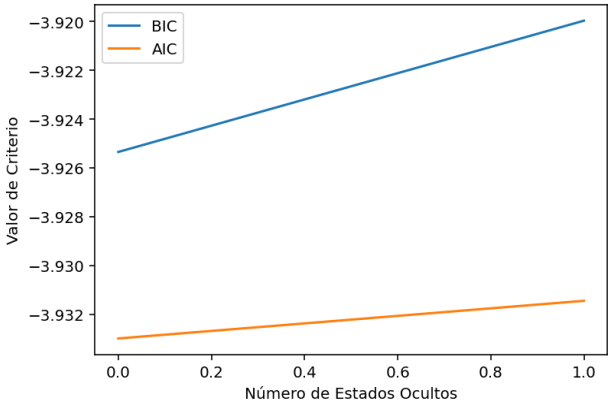

Figura N. 84: Word2Vec-384d, Sph y Texto Lema – Sin Esc

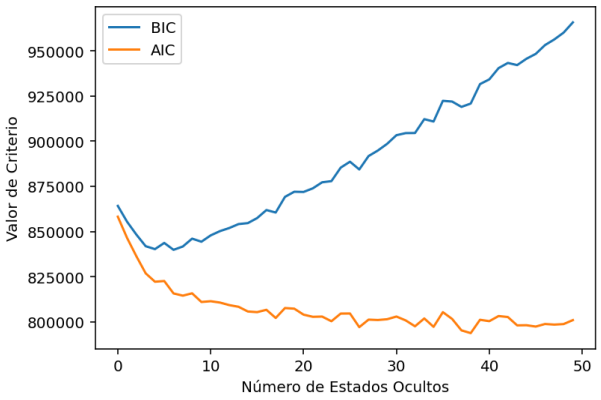

Figura N. 85: GloVe 300d, Sph y Texto Lem –Esc

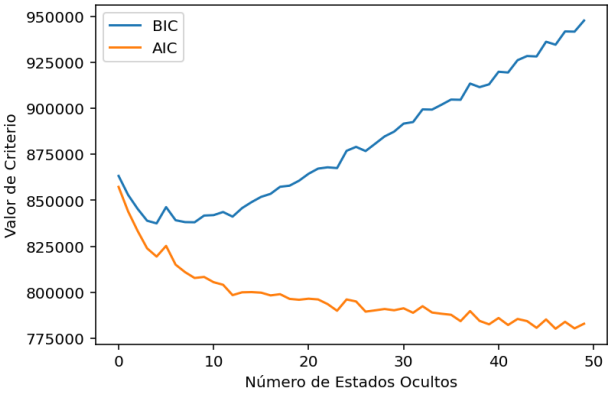

Figura N. 86: GloVe 300d, Diag y Texto Lem –Esc

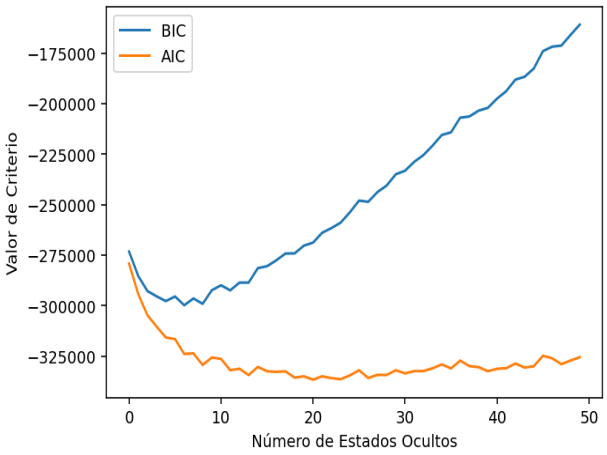

Figura N. 87: GloVe 300d, Sph y Texto Lema –Sin Esc

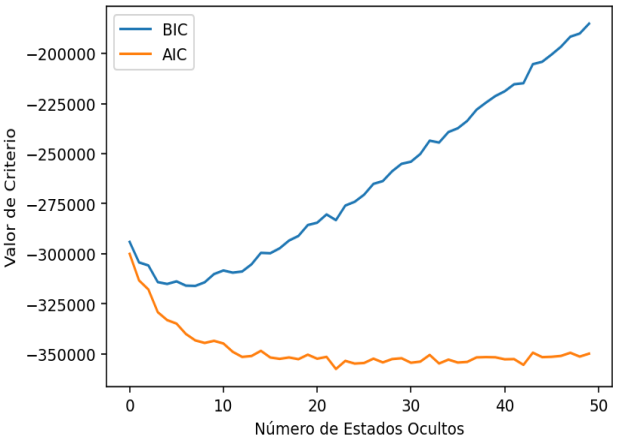

Figura N. 88: GloVe 300d, Diag y Texto Lema –Sin Esc

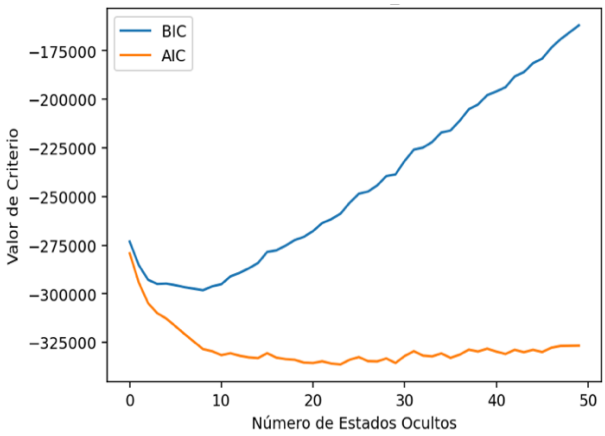

Figura N. 89: GloVe 300d, Sph y Texto POS –Sin Esc

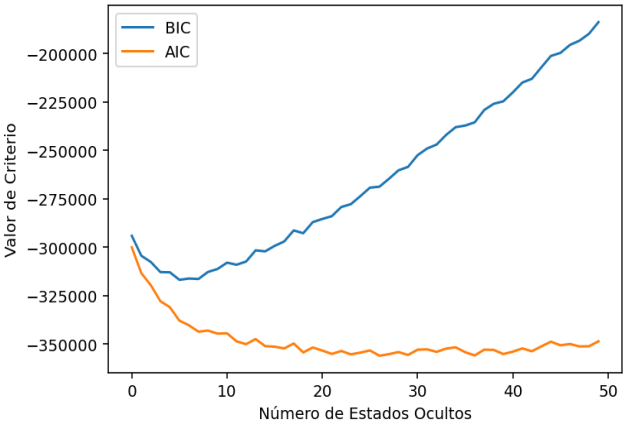

Figura N. 90: GloVe 300d, Diag y Texto POS –Sin Esc

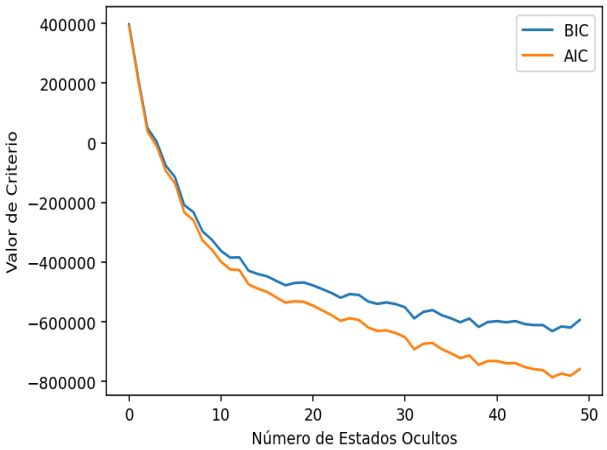

Figura N. 91: FastText-300, Diag y Texto POS –Esc

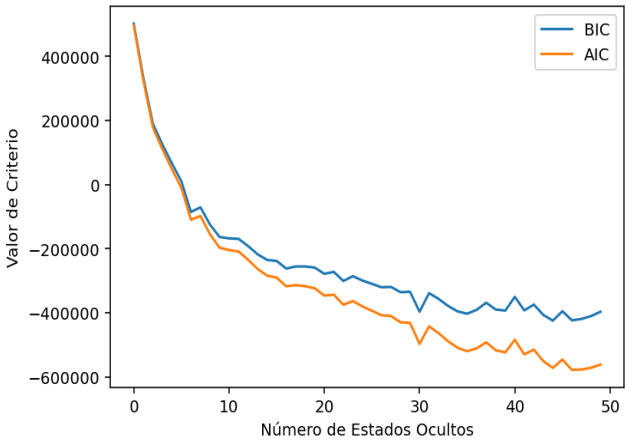

Figura N. 92: FastText-300, Sph y Texto POS –Esc

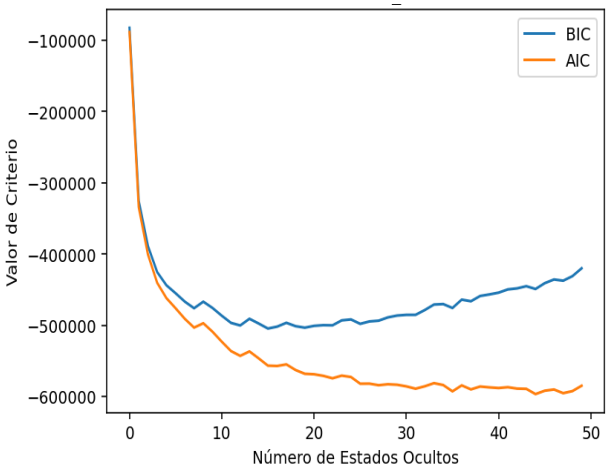

Figura N. 93 FastText-300, Diag y Texto POS –Esc

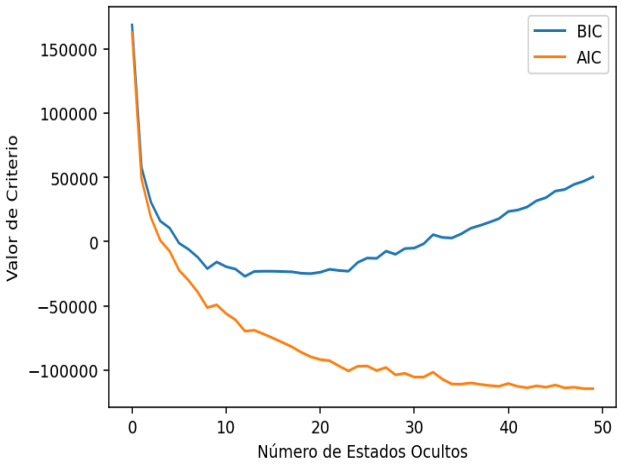

Figura N. 94: FastText-300, Sph y Texto POS –Esc

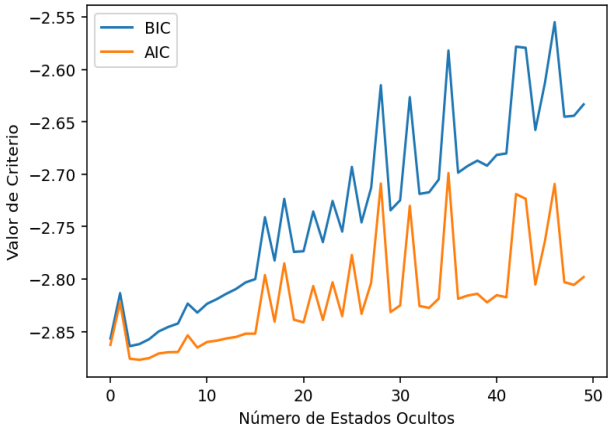

Figura N. 95: FastText-300, Sph y Texto POS –Sin Esc

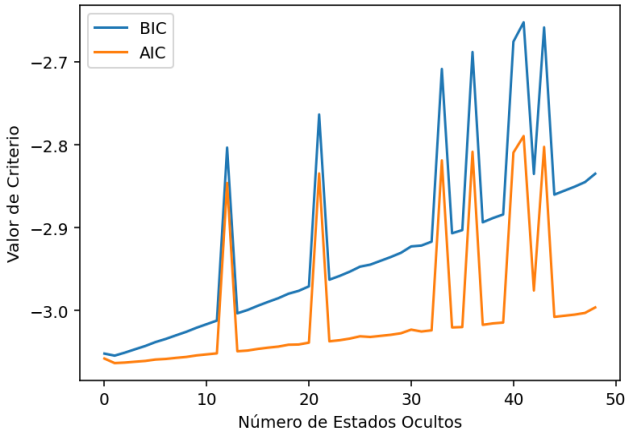

Figura N. 96: FastText-300, Sph y Texto Lem –Sin Esc

Anexo N.3: Resultados adicionales del dataset IMDb

Estados Ocultos	Vectorización	Clasificación POS Tagging	Normalización de Datos	Tipo de Covazianza	Accuracy	F1-Score
39	Embedding	Si	Si	spherical	0.6500	0.6487
35	Embedding	Si	Si	tied	0.6364	0.6371
38	TF-IDF	Si	No	tied	0.6364	0.6364
35	Embedding	Si	Si	diag	0.6364	0.6323
40	Embedding	Si	No	tied	0.6316	0.6204
40	TF-IDF	Si	Si	tied	0.6071	0.6128
19	Bag of Words	Si	No	spherical	0.5862	0.5827
10	Bag of Words	Si	No	spherical	0.5556	0.5546
19	Embedding	No	Si	diag	0.5417	0.5456
21	Embedding	Si	No	full	0.5400	0.5336
10	TF-IDF	No	No	full	0.4754	0.4803
16	TF-IDF	Si	No	full	0.4697	0.4651
32	TF-IDF	Si	Si	full	0.4691	0.4570
7	Embedding	No	Si	diag	0.4563	0.4455
8	Bag of Words	Si	No	full	0.4359	0.4434
6	Embedding	No	Si	diag	0.4384	0.4399
6	Embedding	No	Si	diag	0.4384	0.4399
9	Bag of Words	No	No	diag	0.4524	0.4392
9	Bag of Words	No	No	diag	0.4524	0.4392
35	Embedding	Si	No	spherical	0.4382	0.4373
6	TF-IDF	No	No	diag	0.4416	0.4315
6	TF-IDF	No	No	diag	0.4416	0.4315
5	TF-IDF	No	No	diag	0.4286	0.4227
5	TF-IDF	No	No	diag	0.4286	0.4227
10	Bag of Words	Si	No	tied	0.4219	0.4186
4	Bag of Words	No	No	diag	0.4216	0.4142
4	Bag of Words	No	No	diag	0.4216	0.4142
9	TF-IDF	Si	No	diag	0.4556	0.4136
5	Bag of Words	No	No	spherical	0.4062	0.4018
7	TF-IDF	No	No	tied	0.4096	0.3976
3	TF-IDF	Si	Si	diag	0.4078	0.3929
7	Embedding	Si	Si	full	0.4141	0.3882
7	Bag of Words	No	No	full	0.3981	0.3846

4	Bag of Words	Si	Si	spherical	0.4227	0.3830
8	Bag of Words	Si	Si	diag	0.4211	0.3696
6	Bag of Words	Si	No	diag	0.4222	0.3653
3	Bag of Words	Si	Si	tied	0.4272	0.3555
2-10	Bag of Words	No	No	tied	0.3918	0.3512
9	Bag of Words	No	Si	diag	0.4369	0.3498
10	Bag of Words	Si	Si	full	0.3724	0.3298
5	Embedding	Si	No	diag	0.3737	0.3273
4	Embedding	No	Si	spherical	0.3883	0.3143
3	Embedding	No	No	diag	0.3786	0.3142
3	Embedding	No	No	diag	0.3786	0.3142
10	TF-IDF	No	Si	diag	0.4200	0.3082
10	TF-IDF	No	Si	diag	0.4200	0.3082
3	Embedding	No	No	diag	0.3592	0.2970
3	Embedding	No	No	diag	0.3592	0.2970
10	Bag of Words	No	Si	diag	0.3641	0.2785
10	Bag of Words	No	Si	diag	0.3641	0.2785
4	Bag of Words	No	Si	diag	0.3981	0.2376
4	TF-IDF	No	Si	diag	0.3786	0.2354
4	TF-IDF	No	Si	diag	0.3786	0.2354
2-10	TF-IDF	No	Si	full	0.3883	0.2173
2-10	TF-IDF	No	Si	tied	0.3883	0.2173
2-10	Bag of Words	No	Si	spherical	0.3689	0.1989
2-10	Bag of Words	No	Si	full	0.3689	0.1989
2-10	Bag of Words	No	Si	tied	0.3689	0.1989
2-10	Embedding	No	Si	full	0.3689	0.1989
47	Word2Vec-384d	Si	Si	diag	0,5000	0,8833
42	FastText-300	Si	Si	diag	0,6880	0,6961
42	BERT-base	Si	Si	diag	0,6818	0,6961
45	GloVe 300d	Si	No	tied	0,4081	0,6775
45	GloVe 300d	Si	Si	tied	0,4081	0,6775
46	FastText-300	Si	Si	spherical	0,5416	0,6751
46	BERT-base	Si	Si	spherical	0,5416	0,6751
49	Word2Vec-384d	Si	Si	spherical	0,6111	0,6692
26	GloVe 300d	No	Si	spherical	0,5121	0,6501
19	GloVe 300d	No	No	spherical	0,5145	0,6225
29	FastText-300	No	No	full	0,6451	0,6108
44	GloVe 300d	No	No	diag	0,5455	0,6097

30	GloVe 300d	No	Si	diag	0,4318	0,5848
26	GloVe 300d	No	No	tied	0,3766	0,5813
50	GloVe 300d	No	No	full	0,3980	0,5715
21	FastText-300	No	Si	diag	0,5000	0,5704
21	BERT-base	No	Si	diag	0,5000	0,5704
25	FastText-300	No	No	tied	0,4489	0,5561
26	FastText-300	No	Si	spherical	0,4324	0,5373
26	BERT-base	No	Si	spherical	0,4324	0,5373
34	Word2Vec-384d	No	Si	diag	0,4909	0,5337
34	GloVe 300d	No	Si	full	0,4197	0,5128
34	Word2Vec-384d	No	Si	spherical	0,4500	0,5113
40	GloVe 300d	No	Si	tied	0,4722	0,501
27	FastText-300	Si	No	full	0,3980	0,4970
27	FastText-300	Si	Si	full	0,3980	0,4970
27	FastText-300	No	Si	full	0,3980	0,4970
31	FastText-300	Si	No	tied	0,4651	0,4758
31	FastText-300	Si	Si	tied	0,4651	0,4758
45	FastText-300	Si	No	spherical	0,3900	0,4129
45	BERT-base	Si	No	spherical	0,3900	0,4129
8	FastText-300	No	No	spherical	0,3883	0,3450
8	BERT-base	No	No	spherical	0,3883	0,3450
43	Word2Vec-384d	Si	No	spherical	0,3883	0,3387
4	FastText-300	Si	No	diag	0,3921	0,2929
4	BERT-base	Si	No	diag	0,3921	0,2929
2	Word2Vec-384d	Si	No	diag	0,3980	0,2377
2	GloVe 300d	Si	Si	spherical	0,3883	0,21726
2	Word2Vec-384d	No	No	spherical	0,3883	0,2172
2	GloVe 300d	Si	No	spherical	0,3883	0,2172
0	Word2Vec-384d	No	No	diag	0,0000	0,0000
0	GloVe 300d	Si	No	diag	0,0000	0,0000
0	GloVe 300d	Si	No	full	0,0000	0,0000
0	GloVe 300d	Si	Si	diag	0,0000	0,0000
0	GloVe 300d	Si	Si	full	0,0000	0,0000
0	FastText-300	No	No	diag	0,0000	0,0000
0	BERT-base	No	No	diag	0,0000	0,0000
0	Word2Vec-384d	Si	No	full	0,0000	0,0000
0	Word2Vec-384d	Si	No	tied	0,0000	0,0000
0	Word2Vec-384d	Si	Si	full	0,0000	0,0000

0	Word2Vec-384d	Si	Si	tied	0,0000	0,0000
0	Word2Vec-384d	No	No	full	0,0000	0,0000
0	Word2Vec-384d	No	No	tied	0,0000	0,0000
0	Word2Vec-384d	No	Si	full	0,0000	0,0000
0	Word2Vec-384d	No	Si	tied	0,0000	0,0000
0	FastText-300	No	Si	tied	0,0000	0,0000
0	BERT-base	Si	No	full	0,0000	0,0000
0	BERT-base	Si	No	tied	0,0000	0,0000
0	BERT-base	Si	Si	full	0,0000	0,0000
0	BERT-base	Si	Si	tied	0,0000	0,0000
0	BERT-base	No	No	full	0,0000	0,0000
0	BERT-base	No	No	tied	0,0000	0,0000
0	BERT-base	No	Si	full	0,0000	0,0000
0	BERT-base	No	Si	tied	0,0000	0,0000

Tabla 10: Resultados adicionales con métricas Accuracy y F1-score del dataset IMDb

Anexo N.4: Resultados adicionales del dataset Yelp

Estados Ocultos	Vectorización	Clasificación POS Tagging	Normalización de Datos	Tipo de Covarianza	Accuracy	F1-Score
44	BERT-base	Si	No	spherical	0.5000	0.7831
44	BERT-base	No	No	spherical	0.5000	0.7831
41	Word2Vec-384d	No	Si	diag	0.5151	0.7548
38	BERT-base	Si	No	diag	0.6842	0.7500
38	BERT-base	No	No	diag	0.6842	0.7500
42	BERT-base	Si	Si	diag	0.5000	0.7472
30	GloVe 300d	No	No	spherical	0.5555	0.7142
24	Word2Vec-384d	Si	Si	spherical	0.6153	0.7095
38	Word2Vec-384d	Si	Si	diag	0.6315	0.7058
23	BERT-base	No	Si	diag	0.5172	0.6956
27	TF-IDF	Si	No	spherical	0.5652	0.6928
22	GloVe 300d	No	Si	spherical	0.5000	0.6848
31	Embedding	Si	Si	spherical	0.4800	0.6835
25	Word2Vec-384d	No	Si	spherical	0.4666	0.6741
27	GloVe 300d	No	Si	diag	0.4523	0.6658
26	GloVe 300d	No	No	diag	0.5000	0.6530
33	Embedding	Si	Si	diag	0.6363	0.6493
45	FastText-300	Si	Si	spherical	0.4666	0.6452
40	TF-IDF	No	Si	diag	0.6129	0.6371
38	Embedding	No	Si	spherical	0.5937	0.6345
25	TF-IDF	No	No	diag	0.6562	0.6342
26	Bow	Si	No	tied	0.4722	0.6275
40	Embedding	No	Si	diag	0.5121	0.5957
41	Bow	Si	Si	spherical	0.4146	0.5901
34	FastText-300	Si	Si	diag	0.5000	0.5898
51	TF-IDF	No	No	spherical	0.5925	0.5881
42	Bow	Si	Si	tied	0.4318	0.5804
51	FastText-300	No	Si	diag	0.5116	0.5317
26	Bow	Si	No	full	0.4583	0.5253
51	Bow	Si	Si	full	0.5142	0.5220
8	Embedding	Si	No	spherical	0.4318	0.5212
41	FastText-300	No	Si	spherical	0.5000	0.5091
27	Bow	No	Si	diag	0.4150	0.4984
4	TF-IDF	No	Si	spherical	0.4945	0.4752
6	TF-IDF	Si	No	diag	0.4526	0.4684
23	TF-IDF	Si	Si	spherical	0.4696	0.4682
12	Bow	Si	No	diag	0.4693	0.4584
6	Bow	No	No	diag	0.4347	0.4498
11	Embedding	No	No	spherical	0.4112	0.4487
15	TF-IDF	Si	Si	diag	0.4700	0.4451
26	FastText-300	Si	No	spherical	0.3750	0.4422
5	Bow	No	No	spherical	0.4020	0.4343

3	Embedding	Si	No	diag	0.3883	0.4330
41	Word2Vec-384d	Si	No	spherical	0.4299	0.4265
4	FastText-300	Si	No	diag	0,4226	0.4257
18	Word2Vec-384d	No	No	spherical	0.4299	0.4138
18	Bow	Si	Si	diag	0.4500	0.4106
4	Embedding	No	No	diag	0.4485	0.4074
3	Bow	No	Si	spherical	0.4299	0.3263
2	Word2Vec-384d	No	No	diag	0.4299	0.2885
2	Word2Vec-384d	Si	No	diag	0.4299	0.2585
2	GloVe 300d	Si	Si	spherical	0.4299	0.2585
2	GloVe 300d	Si	No	spherical	0.4299	0.2585
2	GloVe 300d	Si	No	diag	0.4299	0.2585
2	GloVe 300d	Si	Si	diag	0.4299	0.2585
2	FastText-300	No	No	spherical	0.4299	0.2585
0	FastText-300	No	No	diag	0,0000	0,0000

Tabla 11: Resultados adicionales con métricas Accuracy y F1-score del dataset Yelp