**1. Introducción**

En el presente trabajo se aborda el desarrollo de un sistema inteligente para la clasificación de incidentes en una mesa de servicio utilizando técnicas avanzadas de Procesamiento de Lenguaje Natural (PLN). Se emplean modelos de embeddings para representar los casos históricos y algoritmos de clustering para identificar patrones en los datos. Además, se entrena un modelo supervisado basado en BERT para mejorar la clasificación de incidentes y se desarrolla un chatbot que permite la consulta eficiente de soluciones basadas en casos previos.

**2. Objetivos**

**2.1 Objetivo General**

Desarrollar un chatbot inteligente capaz de clasificar y recomendar soluciones a incidentes en una mesa de servicio, utilizando embeddings y modelos de aprendizaje automático.

**2.2 Objetivos Específicos**

1. Realizar una limpieza y preprocesamiento de los datos históricos de incidentes.
2. Generar embeddings de texto utilizando modelos de representación de palabras.
3. Aplicar algoritmos de clustering para identificar patrones y categorizar los incidentes.
4. Entrenar un modelo supervisado basado en BERT para la clasificación de incidentes.
5. Desarrollar un chatbot que interactúe con los usuarios y proporcione soluciones basadas en los modelos entrenados.
6. Evaluar la precisión del sistema mediante métricas de calidad.

**3. Marco Teórico**

**3.1 Procesamiento de Lenguaje Natural (PLN)**

El PLN es una rama de la inteligencia artificial que permite la interpretación y generación de lenguaje humano. Se fundamenta en modelos de aprendizaje profundo y transformadores como BERT.

**3.2 Embeddings de Texto**

Los embeddings convierten texto en vectores numéricos en un espacio semántico. En este estudio se utilizan:

* **mxbai-embed-large-v1**: Este modelo de embeddings permite representar textos en un espacio vectorial de alta dimensionalidad. Está optimizado para tareas de búsqueda semántica y recuperación de información.

[huggingface.co](https://huggingface.co/mixedbread-ai/mxbai-embed-large-v1?utm_source=chatgpt.com)

* **nomic-embed-text-v1.5**: Nomic es una alternativa optimizada para la representación de texto en tareas de clasificación y clustering.

[nomic.ai](https://www.nomic.ai/blog/posts/nomic-embed-text-v1?utm_source=chatgpt.com)

* **all-MiniLM-L6-v2**: Este modelo es una variante liviana de BERT, diseñado para generar representaciones de texto eficientes con menor costo computacional.

[huggingface.co](https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2?utm_source=chatgpt.com)

Estos modelos permiten representar la semántica de los incidentes y facilitar su clasificación.

**3.3 Algoritmos de Clustering**

Se emplean los siguientes algoritmos para identificar patrones en los incidentes:

* **K-Means**: Es un algoritmo de aprendizaje no supervisado que agrupa datos en k clústeres basándose en la similitud entre ellos. Se evalúa mediante la métrica de Silhouette Score.

[scikit-learn.org](https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html?utm_source=chatgpt.com)

* **DBSCAN**: Este método identifica clústeres basándose en la densidad de los datos, permitiendo encontrar estructuras más complejas.

[scikit-learn.org](https://scikit-learn.org/stable/modules/clustering.html?utm_source=chatgpt.com)

* **HDBSCAN**: Una mejora de DBSCAN que permite identificar clústeres de diferentes densidades sin necesidad de definir un número fijo de grupos.

[medium.com](https://medium.com/%40sangeeth.pogula_25515/clustering-algorithms-in-action-understanding-kmeans-hierarchical-dbscan-and-silhouette-scoring-38c6d89af2c5?utm_source=chatgpt.com)

**3.4 Métricas de Evaluación**

* **Silhouette Score**: Mide la calidad del clustering comparando la cohesión interna de los clústeres con la separación entre ellos.

[scikit-learn.org](https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html?utm_source=chatgpt.com)

* **Davies-Bouldin Index**: Evalúa la separación y compactación de los clústeres generados.

[bioinformatics-training.github.io](https://bioinformatics-training.github.io/intro-machine-learning-2017/clustering.html?utm_source=chatgpt.com)

**3.5 Modelos Supervisados: BERT**

BERT es un modelo basado en transformadores que permite realizar tareas de clasificación de texto con alto grado de precisión. En este trabajo se entrena para categorizar incidentes en 10 clases previamente definidas.

**3.6 Almacenamiento en MongoDB**

MongoDB es una base de datos NoSQL que permite el almacenamiento flexible de documentos JSON. Se utiliza para almacenar los embeddings y los casos históricos para su posterior consulta.

**4. Antecedentes**

Se revisan estudios previos en los que se ha utilizado PLN para la clasificación de incidentes en mesas de servicio. Se destacan trabajos que han empleado modelos de embeddings y clustering para categorizar incidencias y mejorar la resolución de problemas en organizaciones.

**5. Justificación**

El desarrollo de un chatbot inteligente basado en PLN permite reducir tiempos de respuesta y mejorar la eficiencia en la resolución de incidentes. Este sistema contribuirá a optimizar la gestión de la mesa de servicio y mejorar la experiencia del usuario.

**6. Metodología**

Se sigue la metodología CRISP-DM, la cual consta de las siguientes fases:

**6.1 Comprensión del Negocio**

Se analiza la problemática en la mesa de servicio y se definen los objetivos del chatbot.

**6.2 Comprensión de los Datos**

Se realiza un análisis exploratorio de los incidentes históricos para identificar patrones y definir las transformaciones necesarias.

**6.3 Preparación de los Datos**

Se ejecutan procesos de limpieza, tokenización y generación de embeddings para representar los textos en forma numérica.

**6.4 Modelado**

Se implementan:

* Algoritmos de clustering para la identificación de patrones.
* Un modelo basado en BERT para la clasificación supervisada.
* Un chatbot que utilice los modelos anteriores para responder a consultas.

**6.5 Evaluación**

Se utilizan métricas como:

* Coeficiente de silueta para evaluar clustering.
* Precisión, recall y F1-score para medir la clasificación con BERT.

**6.6 Implementación**

Se almacena la información en MongoDB y se despliega el chatbot para pruebas piloto.