Informe Comparativo de Modelos de Clasificación de Mensajes de Soporte

1. Introducción

En este informe se comparan tres enfoques diferentes para la clasificación automática de mensajes de soporte técnico:

- un modelo clásico basado en **TF-IDF con regresión logística**,
- un modelo con **BERT sin fine-tuning**, y
- un modelo con **BERT fine-tuned**, es decir, ajustado específicamente sobre los datos del dominio.

El objetivo principal del análisis es determinar cuál de estos modelos se desempeña mejor en términos de precisión, capacidad de generalización y manejo de clases difíciles, especialmente en contextos de atención al cliente donde una clasificación errónea puede impactar negativamente en los tiempos de respuesta y en la experiencia del usuario.

---

2. Resultados Generales

Los tres modelos fueron evaluados sobre el mismo conjunto de test, compuesto por 5600 ejemplos balanceados entre las clases: **Change**, **Incident**, **Problem** y **Request**. Las métricas utilizadas incluyen *accuracy*, *precision*, *recall* y *F1-score* macro y por clase.

El desempeño global fue el siguiente:

- **TF-IDF + Regresión Logística**:
  Accuracy = 0.90 | F1-score macro = 0.90
- **BERT sin fine-tuning**:
  Accuracy = 0.85 | F1-score macro = 0.85
- **BERT con fine-tuning**:
  Accuracy = 0.97 | F1-score macro = 0.97

Este resumen deja en claro que **el fine-tuning sobre BERT genera una mejora significativa en el rendimiento general del modelo**, superando no solo al modelo clásico sino también a la versión de BERT sin ajustes.

---

3. Análisis por Clase

Un análisis más detallado por clase muestra diferencias clave:

- **Change**: Todos los modelos obtuvieron F1-scores cercanos a 1.00. Se trata de una clase con frases muy específicas y repetitivas (como “I want to change...” o “Please update...”), lo que facilita su detección incluso por modelos simples como TF-IDF.

- **Request**: También se desempeñó muy bien en los tres modelos. El lenguaje formal, estructurado y directo de esta clase (“Can you send me...”, “I need confirmation...”) parece generar embeddings distintivos incluso sin contexto.

- **Incident** y **Problem**: Son las clases más difíciles de diferenciar. Aquí se observaron las mayores diferencias entre los modelos:

    - Con **TF-IDF**, los errores entre Incident y Problem fueron frecuentes. Muchos mensajes de Incident fueron mal clasificados como Problem (307 casos), y viceversa (257 casos).

    - Con **BERT sin fine-tuning**, estos errores incluso aumentaron (405 y 399 respectivamente), lo que sugiere que usar BERT sin entrenamiento específico no solo no mejora el modelo, sino que puede empeorar el desempeño.

    - Con **BERT fine-tuned**, los errores se redujeron a solo 93 (Incident→Problem) y 62 (Problem→Incident), logrando una mejora crítica. Esto demuestra la importancia del fine-tuning para aprovechar el contexto semántico de los mensajes.

---

4. Análisis de Errores

Los errores más comunes se produjeron entre clases que comparten palabras clave pero tienen intenciones diferentes. Por ejemplo, una frase como *“I’m unable to access my account”* podría ser interpretada como un **Incident** o un **Problem**, dependiendo del historial del usuario o del contexto del sistema. Modelos como TF-IDF, que no consideran el orden ni la relación entre palabras, tienden a confundirse en estos casos.

Además, se observó que **BERT sin fine-tuning** no logró aprovechar su arquitectura para reducir estas confusiones. Aunque el embedding de BERT contiene información contextual, si no se ajusta a la tarea específica, los vectores generados no capturan las diferencias sutiles entre clases. En cambio, al hacer fine-tuning, el modelo aprende patrones

específicos de cómo se expresan los problemas técnicos vs. los incidentes reales, lo que resulta en una clasificación mucho más precisa.

---

5. Conclusión

El análisis comparativo muestra de forma concluyente que **el modelo BERT con fine-tuning es el más adecuado para la clasificación automática de mensajes de soporte**. No solo logra un *accuracy* general del 97%, sino que **reduce significativamente los errores críticos entre clases ambigüas**, como *Incident* y *Problem*. Además, mantiene un rendimiento perfecto en las clases más simples como *Change* y *Request*.

Si bien requiere mayor capacidad computacional para el entrenamiento, este modelo es el más recomendable para su implementación en producción, especialmente en entornos donde los tiempos de respuesta, la derivación automática y la priorización de tickets son fundamentales para mejorar la atención al cliente.

Por lo tanto, se concluye que **el fine-tuning no es opcional sino esencial cuando se trabaja con modelos de lenguaje en dominios específicos** como el soporte técnico. Su impacto en la calidad de la clasificación justifica ampliamente el esfuerzo adicional requerido.