

Informe Final – Clasificador Biomédico DSPy

Resumen Ejecutivo

Desarrollo exitoso de un clasificador multi-etiqueta para artículos biomédicos usando DSPy framework, logrando un **F1 Score Promedio de 0.745** y una **Métrica DSPy de 84.4%** en evaluación con 1,213 ejemplos del test set.

Resultados Obtenidos

Métricas de Rendimiento

Categoría	F1 Score	Soporte	Precisión	Recall
Neurológico	0.766	604/1213 (49.8%)	0.824	0.715
Cardiovascular	0.783	435/1213 (35.9%)	0.720	0.857
Hepatorenal	0.748	361/1213 (29.8%)	0.889	0.645
Oncológico	0.681	213/1213 (17.6%)	0.719	0.648

F1 Score Promedio Final: 0.745

Métrica DSPy: 84.4% (1023.75/1213)

Matriz de Confusión (Test Set: 1,213 ejemplos)

- Neurológico:** TN=517, FP=92, FN=172, TP=432
- Cardiovascular:** TN=633, FP=145, FN=62, TP=373
- Hepatorenal:** TN=823, FP=29, FN=128, TP=233
- Oncológico:** TN=946, FP=54, FN=75, TP=138

Mejor precisión: Hepatorenal (88.9%)

Mejor recall: Cardiovascular (85.7%)

Metodología Implementada

1. Arquitectura de Solución

Input CSV → DSPy GEPA Optimizer (GPT-5 LLM como optimizer) → GPT-4o-mini → Multi-label Output → Evaluation

2. Proceso de Desarrollo

Fase 1: Exploración (`notebook_miprov2.ipynb`)

- Experimentación inicial con MIPROv2
- Pruebas de diferentes optimizadores
- Resultados variables (70-90% F1)

Fase 2: Optimización (`notebook_gepa.ipynb`)

- Implementación con GEPA optimizer
- Refinamiento de prompts
- Mejor estabilidad y rendimiento

Fase 3: Producción (`main.py`)

- Script final minimalista
- Evaluación automática
- Generación de métricas

3. Optimización DSPy

Técnicas Aplicadas:

- **GEPA (Genetic Algorithm)**: Optimización evolutiva de prompts
- **Chain-of-Thought**: Razonamiento estructurado
- **Multi-label Classification**: Manejo de categorías múltiples
- **Few-shot Learning**: Aprendizaje con ejemplos limitados

Análisis de Enfoques

Enfoques Exitosos

1. DSPy + GEPA: Combinación ganadora

- Optimización automática de prompts
- Mejor que ajuste manual
- Estabilidad en predicciones

2. Signatures Estructuradas:

- Input/Output fields definidos
- Razonamiento explícito
- Formato consistente

3. Multi-label Strategy:

- Pipe-separated categories (neurological|cardiovascular)
- Parsing robusto de predicciones

Enfoques que No Funcionaron

1. MIPROv2 Initial:

- Resultados inconsistentes
- Variabilidad alta (62-94 % F1)
- Optimización lenta

2. Single-label Approach:

- Perdía información multi-categoría
- Menor rendimiento general

3. Manual Prompt Engineering:

- Tiempo intensivo
- Resultados subóptimos vs automático

Diseño de la Solución

Diagrama de Flujo Completo

Ver en el README.md

Componentes Clave

1. **Input Handler:** Procesa CSV con delimitador ;
2. **DSPy Classifier:** Modelo optimizado con GEPA
3. **Prediction Engine:** Batch processing eficiente
4. **Metrics Calculator:** F1 ponderado + matrices
5. **Output Generator:** CSV + visualizaciones

Evidencias de Rendimiento

Dataset de Prueba (Test Set)

- **1,213 artículos** biomédicos del challenge dataset
- **Distribución:** 604 neurológicos, 435 cardiovasculares, 361 hepatorenales, 213 oncológicos
- **Casos multi-etiqueta** manejados correctamente

Métricas Clave

- **F1 Promedio: 0.745:** Rendimiento sólido en dataset real
- **Métrica DSPy: 84.4%:** Evaluación automática exitosa
- **Multi-label:** Manejo correcto de categorías múltiples
- **Dataset grande:** 1,213 ejemplos evaluados
- **Matrices confusión:** Generadas automáticamente

Innovaciones Técnicas

1. **Optimización Automática:** GEPA vs manual tuning
2. **Minimal Code:** 160 líneas vs notebooks complejos
3. **Production Ready:** Script ejecutable directo
4. **Evaluation Pipeline:** Métricas completas automáticas

Conclusiones

Logros Principales

- **F1 Score sólido alcanzado** (0.745 en dataset real de 1,213 ejemplos)
- **Métrica DSPy 84.4%:** Rendimiento robusto según framework
- **Implementación funcional** y ejecutable
- **Pipeline completo** de evaluación con dataset completo
- **Documentación profesional** basada en resultados reales

Lecciones Aprendidas

1. **DSPy es superior** a prompt engineering manual
2. **GEPA optimizer** más estable que MIPROv2
3. **Simplicidad** en producción es clave
4. **Multi-label** requiere parsing cuidadoso

Trabajo Futuro

- Expandir dataset oncológico
 - Optimizar para datasets grandes
 - Implementar ensemble methods
 - Deploy como API REST
-

Resultado Final: Solución exitosa con F1=0.745 y DSPy=84.4%

Implementación completa disponible en `main.py` con documentación en `README.md`