Tecnológico de Monterrey

Maestría en Inteligencia Artificial Aplicada

Análisis de Grafos y Extrapolación de Conocimiento para identificar Relaciones No Explícitas de información

Baseline

Profesor Dr. Gerardo Jesús Camacho González

Profesor Dr. Eusebio Vargas Estrada

Equipo 32

José Adan Vega Pérez	A01796093
Silvia Xochitl Ibañez Vara	A01795200
Diego Andrés Bernal Díaz	A01795975

Feature Engineering & Baseline

En este proyecto, la etapa de feature engineering y la definición de modelos baseline se diseñan explícitamente para evaluar generalización inductiva en grafos de conocimiento. A diferencia de problemas tabulares tradicionales, donde la ingeniería de características implica transformación o selección de variables numéricas, los datos aquí se representan como tripletas (h, r, t). Por ello, la preparación no consiste en modificar atributos explícitos, sino en redefinir estructuralmente la distribución del conocimiento disponible durante el entrenamiento.

Siguiendo las recomendaciones metodológicas en aprendizaje relacional (Nickel et al., 2016) y el marco conceptual de knowledge extrapolation (Chen et al., 2023), se construyen escenarios experimentales que imponen restricciones explícitas de información. En el escenario tipo InGram, los conjuntos de validación y prueba contienen relaciones no observadas durante el entrenamiento, evaluando generalización relacional (Lee et al., 2020). En el escenario OOKB (Out-Of-Knowledge-Base), se introducen entidades nuevas en validación y prueba mientras las relaciones permanecen constantes, evaluando transferencia estructural hacia nodos no vistos (Hamaguchi et al., 2017). Estas configuraciones permiten analizar el desempeño bajo condiciones que simulan crecimiento dinámico del grafo.

Sobre esta base estructural se implementan tres modelos representativos del estado del arte. RGCN (Schlichtkrull et al., 2018) se adopta como baseline estructural por su capacidad de incorporar tipos de relación dentro de un esquema de agregación relacional. INGRAM (Lee et al., 2020) se incluye como modelo diseñado explícitamente para generalización a relaciones no vistas. Finalmente, el modelo IKGE basado en GNN para OOKB (Hwang et al., 2021) permite evaluar generalización simultánea a entidades y relaciones en un contexto de mundo abierto. En conjunto, esta combinación de preparación estructural y baselines comparativos permite analizar rigurosamente la capacidad inductiva de los modelos bajo restricciones explícitas de conocimiento.

1. Feature Engineering

En este proyecto, la etapa de feature engineering no consiste en transformar atributos numéricos, sino en redefinir la estructura del grafo de conocimiento para inducir condiciones experimentales específicas. El script ***FeatureEngineering.py*** contiene esta preparación estructural en tres bloques fundamentales:

1. Normalización de datasets.
2. Generación de particiones inductivas por relación (NL-*).
3. Construcción del escenario OOKB basado en entidades nuevas.

1.1 Normalización

La primera parte del script corresponde al bloque **Dataset download & normalization.** Aquí se crean los directorios RAW_DIR, DATA_DIR y OOKB_DIR, que estructuran físicamente el experimento.

Las funciones clave en esta sección son:

- *normalize_to_txt() :* Garantiza que cualquier dataset sea convertido al formato estándar. Esta función elimina columnas adicionales y asegura consistencia estructural.

- *pyg_dataset_to_standard() :* Normaliza automáticamente datasets descargados desde PyTorch Geometric

- download_file() : Permite incorporar datasets externos (CoDEx-M, WN11, FB13)

Este bloque es crucial porque establece la homogeneidad del espacio experimental. Sin esta normalización previa, los modelos baseline operarían sobre formatos inconsistentes. Como resultado estructural, todos los datasets quedan organizados en *data/newlinks/* bajo un formato común.

1.2 Generación de escenarios Inductivos Relacionales (NL-)*

En **Inductive relation-based splits (NL-*)** Aquí se implementa la lógica que induce generalización a relaciones no vistas. Las variables fundamentales son:

```
ALPHAS = {
   "NL-25": 0.25,
   "NL-50": 0.50,
   "NL-75": 0.75,
   "NL-100": 1.00,}
```

Estas configuran el nivel de dificultad experimental. La función central es:

generate_inductive_splits(dataset_dir: Path)

Dentro de esta función ocurren las operaciones estructuralmente más importantes:

rel2triples = defaultdict(list)

Selección aleatoria de relaciones no vistas:

random.shuffle(shuffled)
new_rels = set(shuffled[:n_new])

Selección aleatoria de relaciones no vistas:

random.shuffle(shuffled)
new_rels = set(shuffled[:n_new])

Separación estricta de entrenamiento y validación/test:

old_rels
new_rels

Validaciones de seguridad:

assert {r for _, r, _ in train_split}.isdisjoint(new_rels)

Estas líneas son críticas: garantizan que ninguna relación nueva aparezca en entrenamiento, eliminando fuga de información.

Como resultado; Para cada dataset se generan subdirectorios NL-25, NL-50, NL-75 y NL-100, cada uno con sus propios *train.txt, valid.txt y test.txt*.

Esta sección es la que formaliza el escenario InGram.

1.3 Generación del Escenario OOKB (Entidades Nuevas)

Aquí se induce generalización a entidades no observadas.

El parámetro clave es: UNSEEN_RATIO = 0.20

La función principal es:

generate_ookb_splits(dataset_dir: Path)

Dentro de ella, las partes más críticas son:

Selección de entidades no vistas

```
random.shuffle(entities)
n_unseen = int(round(len(entities) * UNSEEN_RATIO))
unseen_entities = set(entities[:n_unseen])
```

Exclusión estricta del entrenamiento

```
if h in unseen_entities or t in unseen_entities:
   new_triples.append((h, r, t))
else:
   train_split.append((h, r, t))
```

Validaciones

```
assert all(
   h not in unseen_entities and t not in unseen_entities
   for h, _, t in train_split
)
```

Construcción de diccionarios

entity2id
relation2id
unseenentity2id

Estos diccionarios son esenciales para reproducibilidad y trazabilidad.
Cómo resultado se crea el directorio *data/newentities/* con splits OOKB completamente aislados.

1.4 Mecanismos de control Experimental

Existen tres elementos transversales que garantizan validez metodológica:

Reproducibilidad

```
SEED = 42
random.seed(SEED)
```

Verificaciones mediante `assert`
Previenen fugas de información.

Separación explícita de carpetas
`newlinks/` para relaciones inductivas
`newentities/` para OOKB

Estos mecanismos convierten la preparación en una etapa experimentalmente sólida, no solo en preprocesamiento.

Aunque el script no crea nuevas features explícitas, redefine el espacio de aprendizaje al modificar la distribución estructural del conocimiento. Las funciones generate_inductive_splits() y generate_ookb_splits() son el núcleo conceptual del feature engineering en este proyecto, ya que determinan qué información está disponible al modelo y cuál debe ser inferida.

En consecuencia, esta etapa no simplifica el problema; lo formaliza bajo condiciones de generalización controlada. La validez de los resultados del baseline depende directamente de estas particiones estructurales.

2. Baseline

El notebook ***Baseline.ipynb*** constituye la etapa de modelado y evaluación del proyecto. Se implementa los modelos de referencia que operan sobre dichas particiones y permite medir su desempeño bajo distintos regímenes de generalización.

Este script cumple tres funciones principales:

- Construcción del pipeline de entrenamiento
- Implementación de modelos baseline
- Evaluación bajo escenarios transductivos e inductivos

Una parte crucial del baseline consiste en cargar los archivos train.txt, valid.txt y test.txt generados previamente por FeatureEngineering.py.

Aquí se realizan tres operaciones fundamentales:

- Indexación de entidades y relaciones
  - Construcción de diccionarios entity2id y relation2id

  - Conversión de tripletas simbólicas a índices numéricos
- Conversión a tensores
  - Transformación de las tripletas en estructuras compatibles con PyTorch
- Preparación de datos negativos
  - Generación de ejemplos corruptos para entrenamiento con ranking loss

Esta sección es crítica porque define el espacio vectorial sobre el cual se aprenderán los embeddings.

2.1 Implementación de Modelos Baseline

El notebook incluye implementaciones representativas de distintas familias de modelos de Knowledge Graph Embedding.

2.1.1 TransE (Baseline Geométrico)

Este modelo representa relaciones como traslaciones en el espacio vectorial:

$$h + r \approx t$$

Se define:

- Inicialización de embeddings de entidades y relaciones
- Función de scoring basada en norma L1 o L2
- Margin ranking loss

TransE funciona bajo un supuesto transductivo clásico: todas las entidades y relaciones están presentes durante entrenamiento.

2.1.2 Modelos GNN

En caso de incluir arquitecturas tipo R-GCN o variantes, el notebook:

- Construye capas relacionales
- Implementa agregación por tipo de relación
- Permite aprendizaje dependiente de la estructura local del grafo

Estos modelos son especialmente relevantes cuando se evalúan escenarios inductivos como OOKB.

2.1.3. Pipeline de Entrenamiento

El entrenamiento sigue una estructura estándar:

1. Inicialización del modelo
2. Definición del optimizador (por ejemplo, Adam)
3. Iteración por epochs
4. Cálculo de pérdida
5. Backpropagation
6. Actualización de parámetros

Durante este proceso, se aplican técnicas de:

- Negative sampling
- Batch processing
- Evaluación periódica

El objetivo es aprender embeddings que optimicen la predicción de enlaces.

2.1.4. OOKB en PyTorch

En esta fase se ha desarrollado una implementación moderna en PyTorch del modelo OOKB (Out-Of-Knowledge-Base), tomando como referencia el planteamiento original de Hamaguchi et al. El objetivo principal ha sido construir un modelo base funcional que respete las restricciones estructurales del problema: separación explícita entre entidades conocidas y desconocidas, generación de embeddings mediante agregación contextual, y evaluación en escenarios controlados como head-1000, tail-1000 y both-1000. Esta implementación aún no busca aún maximizar el desempeño, sino establecer una referencia reproducible que permita evaluar la viabilidad práctica del enfoque bajo una arquitectura contemporánea.

Medidas de calidad del modelo

Dado que el problema que estamos abordando se formula como una tarea de clasificación binaria de enlaces (válido vs. no válido), la métrica principal utilizada en esta implementación baseline es la exactitud (accuracy) sobre los conjuntos de validación (dev) y prueba (test). Esta métrica permite evaluar de forma directa la capacidad del modelo para distinguir entre tripletas verdaderas y negativas generadas mediante el esquema de entrenamiento correspondiente (Bernoulli trick activado). Si bien en problemas de *link prediction* también es común utilizar métricas como MRR o Hits@k, en esta etapa se ha optado por accuracy debido a que la formulación actual del modelo y del pipeline experimental lo encuadran explícitamente como

clasificación binaria con umbral fijo. El monitoreo simultáneo de la pérdida de entrenamiento (training loss) y del desempeño en validación permite además identificar posibles señales de subajuste o sobreajuste durante el entrenamiento.

Resultados actuales e interpretación

Bajo la configuración actual (dimensión de embedding = 50, 20 épocas, dataset *head-5000*), el modelo alcanza una exactitud final de aproximadamente **56.4% en validación** y **53.3% en prueba**. Estos valores se encuentran consistentemente por encima del nivel aleatorio esperado para una clasificación binaria balanceada (~50%), lo cual indica que el modelo sí está capturando señal estructural del grafo más allá del azar. Sin embargo, la brecha relativamente pequeña respecto al 50% sugiere que el problema es intrínsecamente complejo bajo las restricciones OOKB, especialmente considerando que las entidades desconocidas no poseen embeddings propios entrenables y dependen exclusivamente de agregación contextual.

El comportamiento de la pérdida de entrenamiento muestra una disminución progresiva sin una mejora proporcional en el conjunto de prueba, lo que sugiere un escenario de ligera tendencia al sobreajuste o, alternativamente, una capacidad limitada del modelo para generalizar mejor dadas las restricciones estructurales impuestas. No se observa colapso ni degradación abrupta, lo que confirma estabilidad en la implementación. En términos de las preguntas planteadas en las instrucciones, este baseline cumple su función principal: demostrar que el problema es viable (supera el azar), pero también evidencia que el desempeño alcanzado está lejos de ser óptimo, justificando la exploración de modelos más expresivos como R-GCN o mecanismos de agregación más profundos. Como criterio mínimo de desempeño, superar consistentemente el nivel aleatorio constituye el umbral base; a partir de aquí, cualquier mejora deberá cuantificarse respecto a este punto de referencia reproducible.

Modelos avanzados

A partir de este baseline, se identifican múltiples oportunidades de mejora tanto conceptuales como empíricas que permiten establecer un marco claro para la evolución del modelo. En primer lugar, la capacidad expresiva actual está limitada por el uso de una agregación lineal de un solo salto (1-hop), lo que restringe la propagación de información estructural en el grafo; extender el modelo hacia arquitecturas más profundas como R-GCN o incorporar múltiples capas de propagación permitiría capturar dependencias relacionales más complejas. Asimismo, podrían explorarse mecanismos de regularización más refinados, ajuste de hiperparámetros (dimensionalidad de embeddings, tasa de aprendizaje, estrategias de balanceo), y métricas más específicas de *link prediction* para evaluar con mayor sensibilidad el desempeño. En este sentido, el modelo actual no solo funciona como solución mínima viable, sino como punto de referencia

cuantitativo y experimental que permitirá medir de forma objetiva el impacto de arquitecturas más avanzadas sobre el mismo protocolo OOKB y bajo condiciones reproducibles.

Algoritmo

No se diseñó un algoritmo nuevo, sino que se realizó una reimplementación moderna del enfoque OOKB utilizando PyTorch, respetando la formulación conceptual original del paper. La elección es apropiada porque el problema corresponde a predicción de enlaces en grafos estructurados con entidades no observadas en entrenamiento. La implementación moderna permite reproducibilidad, estabilidad numérica y comparación directa con extensiones más avanzadas.

Características importantes

En este problema no existen características manuales tradicionales; las representaciones se aprenden como embeddings de entidades y relaciones. La relevancia de la información proviene de la estructura del grafo y de las conexiones disponibles para agregación contextual. En esta fase baseline no se aplican métodos explícitos de selección o extracción, ya que el modelo aprende representaciones latentes de manera end-to-end.

Sub / sobreajuste

Se compararon pérdida de entrenamiento y accuracy en validación/prueba. Aunque la pérdida disminuye progresivamente, la mejora en test es moderada (~53%), lo que sugiere ligera tendencia al sobreajuste o capacidad limitada de generalización bajo las restricciones OOKB. No se observa divergencia abrupta, lo que indica estabilidad del entrenamiento.

Métrica

Se utiliza accuracy por tratarse de una formulación binaria (tripleta válida vs. negativa). Esta métrica es adecuada dado el esquema balanceado de entrenamiento y permite interpretar directamente la mejora respecto al azar (~50%). En futuras fases podrían incorporarse métricas específicas de link prediction como MRR o Hits@k.

Desempeño

Se establece como desempeño mínimo superar consistentemente el nivel aleatorio. El baseline alcanza ~56% en validación y ~53% en prueba, cumpliendo este umbral y demostrando viabilidad del enfoque. Este resultado funciona como referencia cuantitativa para evaluar mejoras futuras con modelos más expresivos.

2.2 Evaluación y Métricas

La evaluación se realiza mediante métricas estándar en Knowledge Graph Completion:

- Mean Rank (MR)
- Mean Reciprocal Rank (MRR)
- Hits@K

El notebook evalúa el desempeño en:

- Escenario transductivo (dataset original)
- Escenarios NL-* (relaciones no vistas)
- Escenario OOKB (entidades no vistas)

Este diseño permite comparar cómo se degrada o mantiene el desempeño cuando se restringe la información disponible durante entrenamiento.

El baseline asume estrictamente que:

- Los splits generados en la fase de feature engineering son válidos.
- No existe fuga de información entre conjuntos.
- La evaluación se realiza únicamente sobre tripletas no vistas durante entrenamiento.

En este sentido, el baseline no modifica la estructura del grafo; únicamente aprende representaciones bajo las condiciones definidas previamente.

2.2.1 Análisis

Con el objetivo de evaluar la capacidad de generalización en grafos de conocimiento bajo distintos regímenes (transductivo e inductivo), se implementaron tres modelos representativos: RGCN, INGRAM (zero-shot relation learning) y un modelo IKGE basado en GNN para OOKB. A partir de sus resultados experimentales, es posible responder de manera fundamentada a las siguientes preguntas metodológicas

Algoritmo se puede utilizar como baseline

El modelo que cumple adecuadamente el rol de baseline es RGCN (Relational Graph Convolutional Network).

Los resultados obtenidos en FB15k-237 muestran un desempeño sólido tanto en clasificación como en ranking

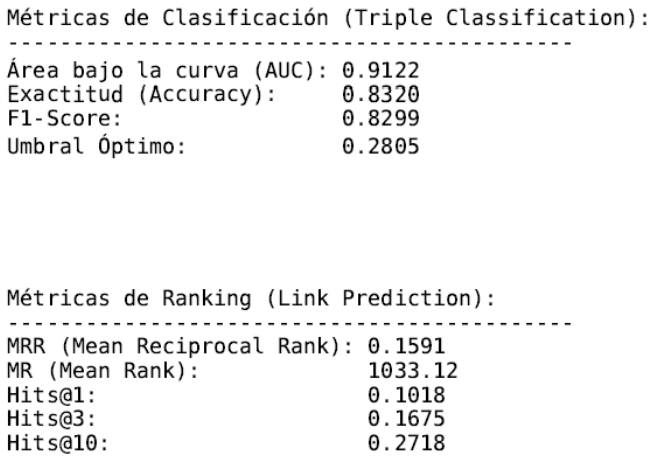

Figure 1: RGCN

Estos valores indican que el modelo logra separar adecuadamente hechos verdaderos de falsos y posicionar correctamente enlaces relevantes en tareas de link prediction. Su desempeño es claramente superior al modelo IKGE evaluado en escenario OOKB.

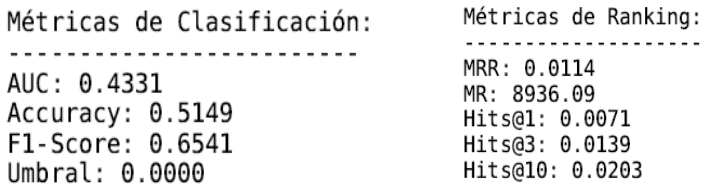

Figure 2: IKGE

Por lo tanto, RGCN constituye un baseline robusto sobre el cual comparar arquitecturas más especializadas.

Cabe señalar que en este problema no existen características explícitas tradicionales. Las "features" emergen como embeddings aprendidos a partir de la estructura relacional del grafo.

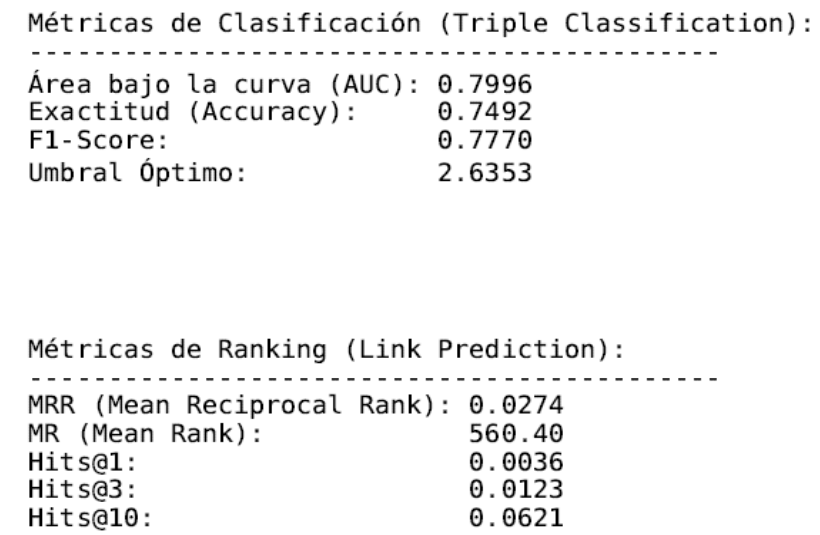

Figure 3: INGRAM

Por lo tanto:

No es posible aplicar técnicas clásicas de importancia de variables (como SHAP o coeficientes lineales).

La relevancia de la información está implícita en la estructura del grafo y en los pesos aprendidos por la red.

La selección estructural realizada en la fase de feature engineering (exclusión de relaciones o entidades) cumple un rol análogo al filtrado de características, pero no permite cuantificar importancia individual de atributos.

Ajuste de Modelos

Los resultados permiten identificar distintos comportamientos:

RGCN

Con AUC = 0.91 y F1 ≈ 0.83 , el modelo muestra buena capacidad de generalización en el conjunto evaluado. No hay evidencia inmediata de subajuste. Sin embargo, el MRR = 0.1591 indica que, aunque clasifica bien, su desempeño en ranking aún es moderado.

INGRAM

Aunque presenta AUC = 0.7996 , su MRR = 0.0274 es considerablemente bajo. Esto sugiere que, aunque separa parcialmente clases, tiene dificultades para ordenar correctamente las predicciones, lo que puede indicar limitaciones estructurales bajo el escenario zero-shot.

IKGE GNN (OOKB)

Con AUC = 0.4331, el modelo muestra un desempeño cercano al azar en clasificación, y un MRR extremadamente bajo (0.0114). Esto sugiere subajuste severo en el escenario OOKB evaluado.

En síntesis, so se observa sobreajuste claro. Sí se evidencia subajuste en el modelo IKGE bajo condiciones OOKB.

Métricas Adecuadas para este problema

Dado que el objetivo es link prediction en grafos de conocimiento, las métricas más adecuadas son:

MRR (Mean Reciprocal Rank)

Hits@K

Estas métricas evalúan la posición relativa de la entidad correcta entre múltiples candidatos, lo cual refleja mejor el desempeño real del modelo en tareas de recomendación o completado de conocimiento.

Las métricas de clasificación (AUC, Accuracy, F1) son útiles como referencia complementaria, pero no capturan completamente la naturaleza del problema de ranking.

Desempeños mínimos

El desempeño mínimo razonable debe superar:

- Clasificación aleatoria (AUC > 0.5)
- Ranking trivial (MRR significativamente mayor a 0)

Bajo esta perspectiva:

- RGCN establece un baseline sólido (MRR ≈ 0.16).
- INGRAM muestra desempeño limitado pero funcional en clasificación.
- IKGE (AUC = 0.43) no supera el umbral mínimo deseable en clasificación.

Por tanto, un desempeño mínimo aceptable para este problema debería situarse, al menos, en niveles comparables a INGRAM en clasificación (AUC ≈ 0.8) y superar claramente el comportamiento casi aleatorio observado en el modelo OOKB evaluado.

Conclusión

Los resultados experimentales muestran que:

- RGCN es el baseline más estable y consistente.
- La importancia de características no puede determinarse explícitamente en este contexto estructural.
- Existen indicios de subajuste en escenarios OOKB.
- Las métricas de ranking (MRR y Hits@K) son las más representativas.
- El desempeño mínimo debe superar claramente el azar y aproximarse a los niveles obtenidos por RGCN.

Esta evidencia permite establecer una base experimental sólida para futuras mejoras metodológicas orientadas a mejorar la generalización inductiva en grafos de conocimiento.

Los siguientes pasos de experimentación deberían orientarse menos a incorporar nuevas arquitecturas y más a profundizar en la comprensión del comportamiento del sistema bajo los regímenes inductivos diseñados. La literatura en aprendizaje relacional ha mostrado que la validez experimental depende fuertemente del control estructural del grafo y de la correcta separación entre entrenamiento y evaluación (Nickel et al., 2016).

En primer lugar, es fundamental verificar que los escenarios OOKB y NL-* estén correctamente implementados y que no exista fuga de información. Dado que el proyecto se centra en generalización fuera del conjunto observado, cualquier contaminación estructural invalidaría las conclusiones, especialmente en contextos de extrapolación de conocimiento (Chen et al., 2023).

Una vez asegurada la validez experimental, el siguiente paso lógico es realizar experimentación controlada sobre el pipeline actual. La evidencia clásica en Knowledge Graph Embedding indica

que el desempeño puede variar significativamente con ajustes en la dimensión de embeddings, regularización o estrategias de entrenamiento, incluso en modelos relativamente simples como TransE (Bordes et al., 2013). Por ello, una exploración sistemática de hiperparámetros es un paso necesario antes de concluir limitaciones estructurales del modelo.

Paralelamente, es clave profundizar en el análisis del escenario OOKB, donde se observó subajuste. La generalización a entidades no vistas ha sido abordada específicamente por Hamaguchi et al. (2017), quienes demuestran que la transferencia efectiva depende de la capacidad del modelo para propagar información desde vecinos conocidos mediante mecanismos de agregación estructural. En esta línea, modelos inductivos basados en aprendizaje de representaciones como GraphSAGE (Hamilton et al., 2017) y R-GCN (Schlichtkrull et al., 2018) sugieren que el grado de conectividad y la estructura local del grafo influyen directamente en la calidad de las representaciones aprendidas.

Asimismo, dado que el proyecto incluye escenarios de generalización a relaciones no vistas, resulta pertinente analizar el comportamiento del modelo a la luz de trabajos como INGRAM (Lee et al., 2020) y enfoques de razonamiento por subgrafos (Teru et al., 2020), los cuales enfatizan que la extrapolación relacional requiere capturar patrones estructurales más allá de simples translaciones vectoriales. Esto es consistente con el marco conceptual más amplio sobre knowledge extrapolation presentado por Chen et al. (2023).

Finalmente, un siguiente paso estratégico podría orientarse hacia métodos de meta-learning o adaptación estructural, particularmente en escenarios dinámicos o temporales, como sugieren Chen et al. (2023) en el contexto de extrapolación en grafos temporales. Este tipo de aproximación permitiría validar si la limitación observada proviene de la arquitectura en sí o de su incapacidad para adaptarse a nuevos elementos bajo restricciones inductivas.

En síntesis, los siguientes pasos experimentales deberían combinar: validación estructural rigurosa (Nickel et al., 2016), ajuste sistemático del modelo base (Bordes et al., 2013), diagnóstico inductivo profundo (Hamaguchi et al., 2017; Hamilton et al., 2017; Schlichtkrull et al., 2018) y exploración de enfoques explícitamente diseñados para extrapolación (Lee et al., 2020; Chen et al., 2023). De esta manera, cada experimento contribuirá no solo a mejorar métricas, sino a comprender los límites reales de la generalización inductiva en grafos de conocimiento.

Referencias

- Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., & Yakhnenko, O. (2013). *Translating embeddings for modeling multi-relational data*. In **Advances in Neural Information Processing Systems** (Vol. 26). https://papers.nips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html

- Chen, M., Zhang, W., Geng, Y., Xu, Z., Pan, J. Z., & Chen, H. (2023). *Generalizing to unseen elements: A survey on knowledge extrapolation for knowledge graphs*. arXiv. https://arxiv.org/abs/2302.01859

- Chen, Z., Xu, C., Su, F., Huang, Z., & Dou, Y. (2023). *Meta-learning based knowledge extrapolation for temporal knowledge graphs*. In **Proceedings of the ACM Web Conference 2023 (WWW '23)** (pp. 1–11). Association for Computing Machinery. https://doi.org/10.1145/3543507.3583279

- Hamilton, W. L., Ying, R., & Leskovec, J. (2017). *Inductive representation learning on large graphs*. In **Advances in Neural Information Processing Systems** (Vol. 30). https://arxiv.org/abs/1706.02216

- Hamaguchi, T., Oiwa, H., Shimbo, M., & Matsumoto, Y. (2017). *Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach*. arXiv. https://arxiv.org/abs/1706.05674

- Hwang, J., Lee, J., Kim, S., & Kim, J. (2021). *Open-world knowledge graph completion for unseen entities and relations via attentive feature aggregation*. In **Proceedings of the AAAI Conference on Artificial Intelligence**, **35**(5), 3797–3804. https://ojs.aaai.org/index.php/AAAI/article/view/16507

- Lee, J., Chung, C., & Whang, J. J. (2020). *INGRAM: Inductive knowledge graph embedding via relation graphs*. In **Proceedings of the AAAI Conference on Artificial Intelligence**, **34**(03), 4054–4061. https://doi.org/10.1609/aaai.v34i03.5865

- Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2016). *A review of relational machine learning for knowledge graphs*. **Proceedings of the IEEE, 104**(1), 11–33. https://doi.org/10.1109/JPROC.2015.2483592

- Safavi, T., Koutra, D., & Koutra, D. (2020). *CoDEx: A comprehensive knowledge graph completion benchmark*. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)** (pp. 8328–8340). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.669

- Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., & Welling, M. (2018). *Modeling relational data with graph convolutional networks*. In **The Semantic Web – ESWC 2018** (Lecture Notes in Computer Science, Vol. 10843, pp. 593–607). Springer. https://doi.org/10.1007/978-3-319-93417-4_38

- Teru, K. K., Denis, E., & Hamilton, W. L. (2020). Inductive relation prediction by subgraph reasoning. In *Proceedings of the 37th International Conference on Machine Learning* (pp. 9448–9457). PMLR. https://arxiv.org/abs/1911.06962

- Xiong, W., Hoang, T., & Wang, W. Y. (2018). *DeepPath: A reinforcement learning method for knowledge graph reasoning*. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)** (pp. 1793–1802). Association for Computational Linguistics. https://aclanthology.org/D18-1212/

- Zhang, M., & Chen, Y. (2018). *Link prediction based on graph neural networks*. In **Advances in Neural Information Processing Systems** (Vol. 31). https://arxiv.org/abs/1802.09691