

# Node Classification e Link Prediction su Knowledge Graph tramite Graph Neural Network

## Studenti

*Macaluso G. e Mistretta M.*

## Supervisore

*Prof. Pierfrancesco Bellini*

UNIVERSITÀ DEGLI STUDI DI FIRENZE

Scuola di Ingegneria - Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Magistrale in Intelligenza Artificiale

9 March 2022

# Indice

- 1 Introduzione
- 2 Strumenti Utilizzati
- 3 Modelli
- 4 Implementazione
- 5 Valutazione e Comparazione
- 6 Conclusioni

# Obbiettivo

- Implementare una GNN al fine di effettuare Node Classification e Link Prediction su **Km4City**
- Confrontare i risultati ottenuti con quelli di modelli preesistenti

# Packages



Figura: Apache Jena Fuseki



Figura: SPARQLWrapper

# R-GCN

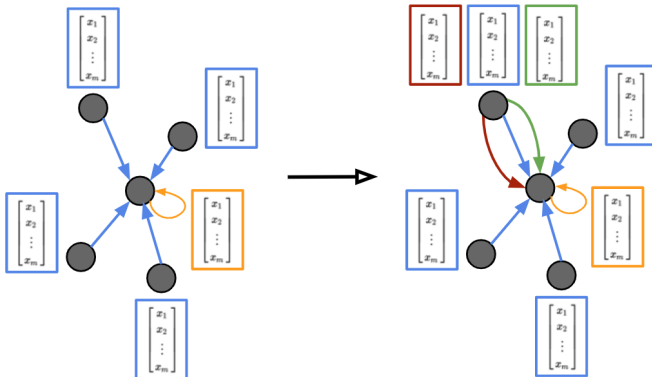


Figura: Modello di funzionamento del message passing

# DQ-GNN

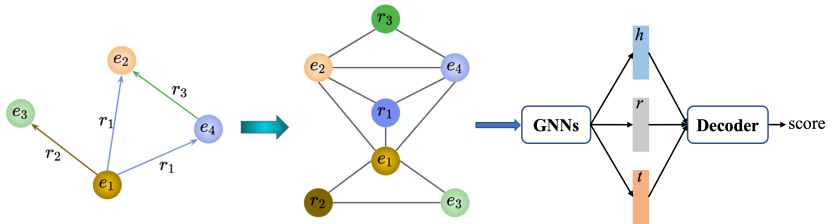
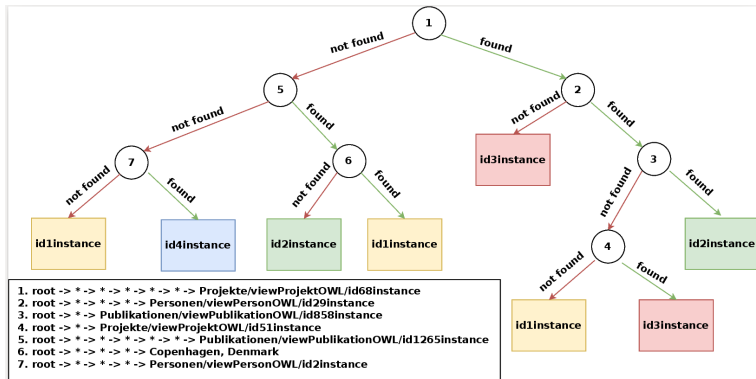


Figura: Modello di funzionamento di Link Prediction con NoGE

# Decision Tree



**Figura:** Modello di funzionamento di Node Classification con MINDWALC

# TuckER

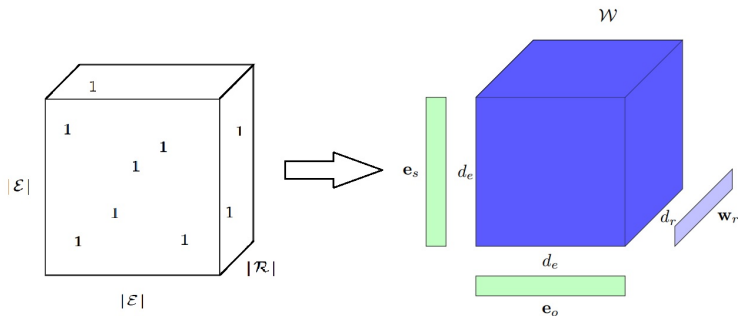


Figura: Modello di funzionamento di Link Prediction con TuckER



# Download Dataset

```
sparql.setQuery("""
PREFIX km4c: <http://www.disit.org/km4city/schema#>
select distinct ?s from <http://www.disit.org/km4city/resource/Eventi/Eventi_a_Firenze> where {
  service <https://servicemap.disit.org/WebAppGrafo/sparql> {?s a km4c:Event }
}
""")
sparql.setReturnFormat(JSON)
risQuery1 = sparql.query().convert()

file = open_secure(file_path, "w")
if want_tsv:
    file.write('s\tp\t\n')

print("\nLista entità km4city:Event ottenuta, avvio ricerca triple a massimo 3 hop di distanza...")

for risQuery1_itr in risQuery1["results"]["bindings"]: # scorro tutti i km4city:Event
    # restituisce tutte le triple a massimo 3 hop di distanza dalla i-esimo evento
    sparql.setQuery("""
PREFIX km4c: <http://www.disit.org/km4city/schema#>
select distinct ?s ?p ?o from
<http://www.disit.org/km4city/resource/Eventi/Eventi_a_Firenze> where {
  service <https://servicemap.disit.org/WebAppGrafo/sparql> {
    {?s ?p ?o.
      FILTER(?s = <"" + str(risQuery1_itr["s"]["value"]) + "">)
      union
      {?s ?p ?o. <"" + str(risQuery1_itr["s"]["value"]) + ""> ?y ?s.}
      union {?s ?p ?o. ?x ?y ?s. <"" + str(risQuery1_itr["s"]["value"]) + ""> ?l ?x.}
    }
  }
}
""")
```

# Reformat Dataset

- Per la **Node Classification** viene creato in automatico un file contenente tutte le triple tranne quelle che fanno riferimento alla categoria delle evento e un file, da suddividere in train e test, che contiene solo le triple che specificano la categoria di ciascun evento
- Per quanto riguarda invece **Link Prediction** invece l'insieme delle triple viene direttamente suddiviso in train e test.

# Risultati

Algoritmo	Accuracy	Tempo
RGCN	72.7	43min
DecisionTree	77.9	7min

**Tabella:** Node Classification su km4city

Algoritmo	Loss	Tempo
DQ-GNN	1.13e-03	2h 6min
TuckER	5.44e-05	2h 44min

**Tabella:** Link Prediction su km4city

## Considerazioni Finali

In conclusione ci riteniamo davvero soddisfatti del lavoro svolto. Dati i risultati ottenuti, ci sentiamo da affermare che l'impiego delle GNN in questo settore è per il momento eccessivamente complessa o poco performante, tuttavia la ricerca è in continuo sviluppo ed è plausibile che in futuro tutto ciò possa cambiare.

Grazie per l'attenzione.