

Sommario

1	Introduzione	3
2	Knowledge Graph	5
2.1	Knowledge Graph e Graph Database.....	6
2.2	Graph Data Model.....	8
2.2.1	RDF Graph.....	9
2.2.2	Property Graph.....	12
2.3	Differenze tra i modelli RDF e Labeled Property Graph.....	14
3	K-MAG, il grafo dei beni artistici e culturali marchigiani.....	17
3.1	Il dominio di interesse e il modello dei dati	17
3.2	Implementazione di K-MAG in Neo4J.....	20
3.3	Incremento del contenuto informativo di K-MAG	23
4	Linguaggi di interrogazione.....	25
4.1	Graph Pattern.....	25
4.2	I linguaggi SPARQL e Cypher	27
4.3	Interrogazioni Cypher e SPARQL a confronto	31
4.4	Tradurre da SPARQL a Cypher: lo stato dell'arte	34
5	Integrazione e analisi del grafo K-MAG	37
5.1	Entity Alignment e integrazione dei dati	38
5.2	Implementazione del processo di integrazione K-MAG	41
5.2.1	Macrofase I: mappatura dei nodi con URI dei Beni Culturali.....	42
5.2.2	Raffinamento del confronto con stop-words e similarità.....	46
5.2.3	Macrofase II: integrazione dati su entità già presenti in K-MAG.....	50
5.2.4	Macrofase III: integrazione dati con nuove entità in K-MAG	54
5.3	Tecniche avanzate per l'integrazione.....	62
5.4	Analisi del grafo K-MAG.....	64
5.4.1	Social Network degli artisti più collegati	66
5.4.2	Localizzazione delle opere e ricerca di un percorso di visita	73
6	Conclusioni e prospettive future.....	78
7	Ringraziamenti.....	80
8	Riferimenti bibliografici e sitografici.....	81
9	Indici	83
9.1	Indice delle figure.....	83
9.2	Indice delle tabelle.....	84

1 Introduzione

Nei diversi contesti che si possono presentare in ambito lavorativo e accademico, è importante identificare metodologie per confrontare e integrare informazioni provenienti da sistemi diversi. In questo ambito i modelli a grafo permettono di connettere in modo uniforme concetti originati con diverse rappresentazioni dei dati per una grande varietà di domini.

Anche se l'espressione Knowledge Graph è in uso da decenni, il significato attuale attribuito da chi si occupa di scienza dell'informazione deriva dall'annuncio nel 2012 del *Google Knowledge Graph*. L'idea fondamentale è quella di utilizzare i grafi, oggetti matematici in grado di modellare sistemi e processi in numerosi campi di applicazione, per rappresentare esplicitamente la conoscenza, e questo implica l'integrazione, la gestione e l'estrazione di informazioni da diverse fonti di dati su larga scala.

I grafi da un lato forniscono un'astrazione concisa e intuitiva in cui i nodi rappresentano le entità e gli archi le relazioni tra di esse. Da un altro lato consentono di posticipare la definizione di uno schema definito in tutti i dettagli, permettendo l'evoluzione dei dati e l'acquisizione di conoscenza incompleta in una modalità più flessibile di quanto sia possibile in un contesto relazionale. I linguaggi di query sui grafi supportano, oltre ai classici operatori relazionali come selezione, proiezione, join e unione, operatori di navigazione per trovare ricorsivamente entità connesse tramite percorsi di lunghezza arbitraria.

I due modelli di grafo più diffusi sono il *Resource Description Framework* (RDF), standard W3C per il Semantic Web arricchito di schemi e ontologie, e il *Labeled Property Graph* (LPG), che permette di avere proprietà sulle relazioni, interrogazioni più immediate ed è adatto per l'attraversamento di grafi complessi. RDF usa un linguaggio standard di query SPARQL, al contrario diversi LPG utilizzano distinti linguaggi di query: uno dei più diffusi è Cypher, supportato dalla comunità Neo4j.

I problemi principali da affrontare nel processo di integrazione tra dati provenienti da sorgenti diverse sono le differenze tra i modelli di grafo, il riconoscimento di duplicati e di entità equivalenti, l'eliminazione delle ridondanze e la separazione di entità distinte con la medesima denominazione. L'approccio Linked Data può consentire di mantenere indipendenti le sorgenti e l'interrogazione congiunta può risultare vantaggiosa rispetto all'importazione fisica delle informazioni.

Questa tesi esplora l'integrazione di dati in un Knowledge Graph partendo da una rappresentazione di informazioni artistiche e culturali relative alla regione storico-geografica marchigiana, definita dall'acronimo K-MAG (Knowledge Marche Art Graph).

Il Capitolo 2 offre una panoramica sui Knowledge Graph, sulle modalità di creazione, gestione e arricchimento degli stessi, e sui diversi modelli di rappresentazione. Viene data una definizione formale dei modelli *RDF Graph* e *Property Graph*, elencando gli aspetti salienti delle due rappresentazioni.

Nel Capitolo 3 si illustra K-MAG, grafo dei beni artistici e archeologici con i relativi autori e con i siti in cui si trovano. Si evidenziano le possibili modalità di integrazione, quali la piattaforma in cui il Ministero della Cultura (MiC) pubblica il proprio patrimonio informativo secondo la logica Linked Open Data.

Il Capitolo 4 è dedicato ai linguaggi di interrogazione. Si illustrano con alcuni esempi concreti le caratteristiche dei linguaggi dichiarativi SPARQL, particolarmente adatto per interrogare Semantic Web e *Linked Data*, e Cypher, più espressivo e dalla sintassi vicina al linguaggio naturale. Viene descritto lo stato dell'arte dell'implementazione della traduzione da SPARQL ai linguaggi di interrogazione *Property Graph*, confrontando la sintassi di analoghe query in SPARQL e Cypher sull'end point Triplestore del Ministero dei Beni Culturali e sul grafo K-MAG caricato sui Graph DB Neo4J.

Nel Capitolo 5 si implementano funzionalità per integrare il Knowledge Graph originale con nuove informazioni e per analizzare in dati del Graph DB. Viene utilizzato il linguaggio Python con librerie che permettono l'uso di query Cypher e query SPARQL e sono esplorati alcuni dei problemi principali che si incontrano in processi di integrazione, in particolare l'Entity Alignment.

Nel Capitolo 6 sono espone le conclusioni e le prospettive future.

2 Knowledge Graph

I grafi sono l'oggetto principale di studio di una branca della matematica (Teoria dei Grafi) e vengono utilizzati nella risoluzione di problemi in numerose aree della conoscenza. Lo studio dei grafi nasce con il *problema dei sette ponti di Königsberg* ("è possibile fare una passeggiata attraversando ogni ponte esattamente una volta e tornare al punto di partenza?"), risolto da Eulero nel 1736 grazie ad una formulazione in termini astratti con punti (nodi) collegati da linee (archi).

Definizione [1]

Un **Grafo** è una coppia $G = (N, E)$ tale che $N \neq \emptyset$, $E = \{\{u, v\} \mid u, v \in N\}$. N è l'insieme dei **nodi** (o *vertici*) del grafo, E è l'insieme degli **archi** (o *lati*, o *spigoli*) del grafo, i cui elementi sono sottoinsiemi di cardinalità 2 di N .

Di solito si considerano grafi con insiemi di nodi finiti. Un grafo è **diretto** (*orientato*) se tutti i suoi nodi sono collegati da archi orientati dal nodo sorgente al nodo destinazione. Se gli stessi nodi del grafo possono essere collegati da più archi si parla di **multigrafo**. Grafi e multigrafi orientati possono avere **cappi**, archi che collegano un nodo con sé stesso. Si chiama **sotto-grafo** un sottoinsieme di nodi di un grafo dato. Per descrivere un grafo, piuttosto che elencare tutti i nodi e gli archi, conviene disegnare un diagramma, la figura 1 descrive e visualizza due grafi diversi che hanno lo stesso insieme N di nodi.

$$N = \{a, b, c, d, e\}, \quad E_{\text{grafo}} = \{\{a, b\}, \{a, c\}, \{a, d\}, \{c, d\}, \{d, e\}\}$$

$$E_{\text{multigrafo diretto}} = \{(a, b), (a, c), (a, c), (a, d), (b, b), (c, d), (d, c), (d, e)\}$$

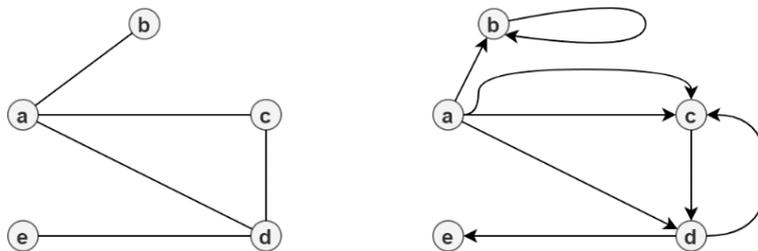


Figura 1 – Grafo e multigrafo diretto con gli insiemi dei nodi e degli archi

2.1 Knowledge Graph e Graph Database

Un Knowledge Graph è un modello di dati che permette di accumulare conoscenza. Il termine è entrato nell'uso comune dal 2012, quando Google lo ha introdotto come funzione sul proprio motore di ricerca e lo ha definito come “un grafo che comprende le entità del mondo reale e le loro relazioni reciproche” [2]. Anche se non totalmente condivisa in letteratura, si adotta la seguente definizione [3]:

Un Knowledge Graph è un grafo di dati destinato ad accumulare e trasmettere la conoscenza del mondo reale, i cui nodi rappresentano entità di interesse e i cui archi rappresentano le relazioni tra queste entità.

Il grafo di dati è conforme ad un modello e la conoscenza, composta da affermazioni semplici (es. “*Ancona è il capoluogo della Regione Marche*”) oppure affermazioni quantificate (es. “*Tutti i capoluoghi di Regione sono città*”), può essere estratta dai dati o accumulata da fonti esterne. Per estrarre ed accumulare ulteriore conoscenza è possibile utilizzare metodi deduttivi (es. “*Ancona è una città*”) oppure induttivi.

I Knowledge Graph sono strumenti utili per analizzare sorgenti di dati grezzi, eterogenei, di grandi dimensioni e in rapida evoluzione. Per supportare la scoperta e l'integrazione dei dati in scenari *data analytics*, sono stati proposti [4] Knowledge Graph in grado di modellare gli indicatori di interesse per un Data Lake semantico.

Il livello di archiviazione di un Knowledge Graph è solitamente un database a grafo. Un Graph Database System [5] è un sistema progettato per la gestione di dati a grafo seguendo i principi di base dei DBMS. Non esiste un modello di dati a grafo unico sul quale possono essere sviluppati tutti i sistemi. La maggior parte dei Graph Database supportano il modello Property Graph, ma possono esserci differenze considerevoli tra i componenti forniti e le loro caratteristiche, in particolare sulla definizione dei linguaggi di query e sull'integrità dei vincoli. Proprio l'assenza di una standardizzazione causa diversi problemi di interoperabilità.

Uno dei maggiori punti di forza dei Graph Database [6] è la capacità di trattare le relazioni tra i dati con una importanza paragonabile ai dati stessi. I Graph DB sono in grado di fornire risposte alle interrogazioni sotto forma di rappresentazioni visuali intuitive ed espressive, sotto-grafi navigabili e dati strutturati esportabili come file. Sono progettati per contenere dati senza limitarli ad un modello fisso e predeterminato e sono efficaci nel gestire query di ricerca complesse.

La flessibilità dei Knowledge Graph permette di partire da un nucleo iniziale, da arricchire in modalità incrementale da altre fonti in base alla necessità e alle possibilità di procurarsi dati. Diversi sono gli approcci possibili [3] per creare e arricchire grafi che contengono conoscenza:

Sorgenti testuali - Testi e corpora sono un'abbondante fonte di dati, ma estrarre informazioni per creare o arricchire un Knowledge Graph senza intermediazione umana è una sfida non banale. Per risolvere questo problema spesso si utilizzano tecniche di Natural Language Processing (NLP) e Information Extraction (IE):

- Le attività di Pre-processing comportano l'applicazione di tecniche sul testo in input (ad esempio tokenizzazione, tagging Part-of-Speech, analisi delle dipendenze, Word Sense Disambiguation).
- La Named Entity Recognition (NER) identifica le entità nominate in un testo, che possono essere utilizzate per generare nuovi nodi candidati per il Knowledge Graph (entità emergenti) o possono essere collegate a nodi esistenti.
- L'Entity Linking (EL) associa le entità nominate in un testo con i nodi esistenti del Knowledge Graph. Questo comporta due problemi da risolvere:
 1. potrebbero esserci più modi per definire la stessa entità, quindi è importante che il Knowledge Graph di destinazione catturi alias ed etichette, anche multilingue, con cui ci si può e ci si deve riferire alla medesima entità;
 2. la stessa citazione in contesti diversi può riferirsi ad entità distinte; pertanto c'è una fase di disambiguazione in cui i candidati vengono classificati e viene scelto dal contesto il nodo più probabile.
- La Relation Extraction (RE) estrae le relazioni tra le entità nel testo: mentre gli approcci tradizionali si basavano su modelli creati manualmente, gli approcci moderni tendono ad utilizzare tecniche di apprendimento supervisionato o non supervisionato.

Sorgenti di tipo mark-up - Il Web è costruito sull'interconnessione di documenti mark-up, nei quali i tag vengono utilizzati per separare gli elementi del documento HTML. Un primo approccio per arricchire un Knowledge Graph estraendo informazioni da documenti mark-up è quello di eliminare i tag, lasciando il solo testo su cui applicare le tecniche NLP/IE. Ma il mark-up può risultare utile per l'estrazione delle informazioni, grazie alle strutture regolari delle pagine di un determinato sito web.

Un approccio tradizionale di estrazione delle informazioni è costituito da *wrapper* definiti manualmente che individuano le informazioni utili direttamente dal documento mark-up. Alcuni approcci si concentrano su specifiche tipologie di mark-up, come le tabelle incorporate nelle pagine web HTML, altri approcci recenti realizzano *wrapper* semi-automatici grazie alla supervisione a distanza e all'Entity Linking.

Deep Web crawling - In questo approccio si cerca di generare input sensati da inviare a moduli Web, input basati su query dell'utente o generati dalla conoscenza del dominio riferimento. I dati vengono estratti dalle risposte con le tecniche viste nei paragrafi precedenti.

Sorgenti strutturate - Gran parte dei dati sono già rappresentati in formati strutturati, in forma tabellare (es. database relazionali e file CSV) o ad albero (es. JSON, e XML). Il caricamento di dati da fonti strutturate prevede due passaggi:

- 1) la creazione di una mappatura dall'origine al grafo;
- 2) l'utilizzo della mappatura per materializzare i dati di origine come grafo o per virtualizzare l'origine come vista grafo sui dati legacy.

2.2 Graph Data Model

Un vantaggio della modellazione dei dati come grafi rispetto al modello relazionale è la possibilità di posticipare la definizione di uno schema, anche se è sempre possibile utilizzare schemi per prescrivere la struttura e/o una semantica di alto livello che il grafo dovrà seguire.

Un **Directed Edge-labeled Graph** è un grafo diretto che contiene un insieme di archi orientati provvisti di una etichetta (figura 2).

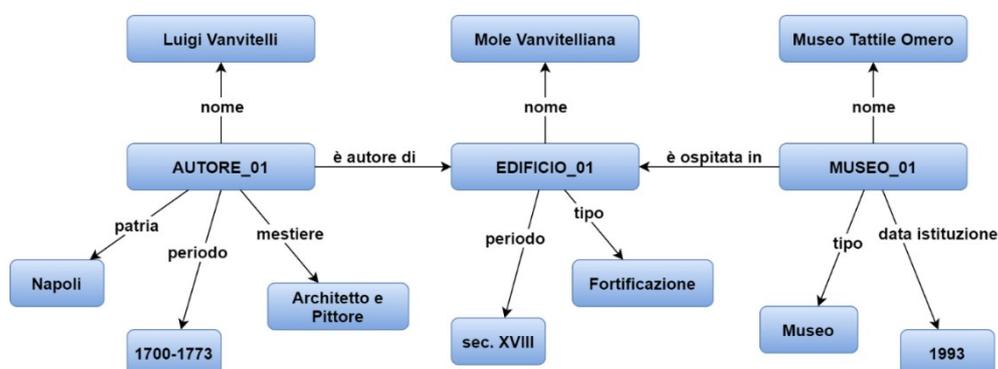


Figura 2 – Esempio di grafo diretto con archi etichettati

Definizione [3]

Un **Directed Edge-labeled Graph** è una tupla $G = (N, E, L)$, dove N è un insieme di nodi, L un insieme di etichette su archi ed $E \subseteq N \times L \times N$ è un insieme di archi.

In figura 2 l'insieme N dei nodi ha 13 elementi, l'insieme E degli archi contiene 12 triple e l'insieme L delle etichette contiene 8 elementi. La definizione non afferma che N e L siano disgiunti, e consente che nodi ed etichette di archi possano essere presenti senza alcun arco associato. In particolari applicazioni possono essere aggiunte restrizioni, restando all'interno della definizione di grafo diretto con archi etichettati.

Nel caso dei Knowledge Graph, i nodi sono utilizzati per rappresentare le entità e gli archi sono utilizzati per rappresentare le relazioni binarie tra tali entità. L'arricchimento del grafo con nuovi dati comporta di solito l'aggiunta di nuovi nodi e/o nuovi archi; le informazioni incomplete sono rappresentate omettendo specifici archi.

2.2.1 RDF Graph

Un modello di dati standardizzato basato su *Directed Edge-labeled Graph* è RDF [3] [7], acronimo di **Resource Description Framework**. È uno standard del World Wide Web Consortium (W3C) creato originariamente per modellare metadati.

Nel modello RDF Triple Stores, la struttura di base è una frase soggetto-predicato-oggetto (figura 3), in cui il predicato è una relazione rappresentata da un arco che collega nodi, mentre il soggetto e l'oggetto sono entità di dati rappresentate dai nodi iniziale e finale.



Figura 3 – RDF triple

Lo standard RDF definisce diversi tipi di nodi:

1. *Internationalized Resource Identifiers* (IRI), identificazione globale delle entità sul Web;
2. *letterali*, che consentono di rappresentare stringhe e altri tipi di dato;
3. *nodi vuoti*, nodi anonimi a cui non è assegnato alcun identificatore.

In un RDF Graph tutto viene chiamato risorsa e non c'è differenza tra le risorse che svolgono un ruolo di arco e le risorse che svolgono un ruolo di nodo. Ogni elemento (soggetto, predicato e oggetto) ha un *Uniform Resource Identifier* (URI) che ne consente l'identificazione in modo univoco.

L'identificatore è un concetto molto importante [9] per RDF Graph e quando due nodi hanno lo stesso identificatore vengono automaticamente considerati come un'unica entità. Gli URI HTTP da un lato possono essere pubblicati sul web ed essere utilizzati pubblicamente, dall'altro è possibile collegare triple RDF al proprio DB Graph, ottenendo ricchi dataset con informazioni provenienti da altre sorgenti.

In figura 4 è illustrato un esempio di RDF Graph relativo all'entità che rappresenta la Mole Vanvitelliana di Ancona nel Catalogo dei Beni Culturali [8] insieme alle risorse di tipo entità e letterali ad essa collegati.

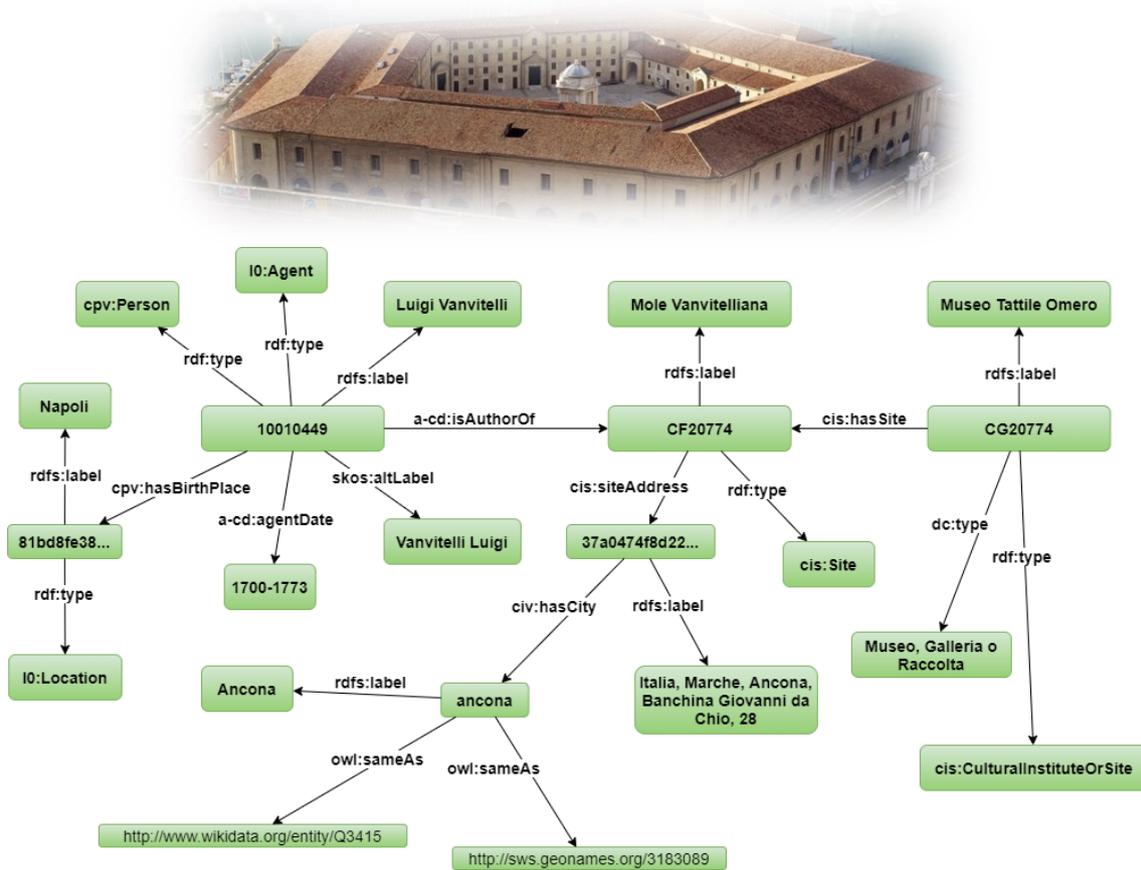


Figura 4 – Esempio di RDF Graph con l'entità che rappresenta la Mole Vanvitelliana

Uno schema semantico consente di definire un vocabolario contenente il significato dei termini di alto livello utilizzati nel grafo. Uno standard importante per definire uno schema semantico per RDF Graph è lo standard RDF Schema (RDFS), che consente di definire sottoclassi, sotto-proprietà, domini e intervalli. In generale la semantica dei termini utilizzati in un grafo viene definita in modo più approfondito con lo standard Web Ontology Language (OWL).

Gli schemi semantici sono tipicamente definiti per Data Graph incompleti e adottano l'Open World Assumption (OWA), quindi l'assenza di un arco tra due nodi non significa che la relazione non possa valere nel mondo reale. Se al contrario si adottasse il Closed World Assumption (CWA), come avviene in molti Data Base System classici, si assumerebbe che il grafo fosse una descrizione completa del mondo, affermando con certezza che non può esistere una relazione tra le due entità. Un compromesso tra OWA e CWA è il Local Closed World Assumption (LCWA), in cui si presuppone che parti del Data Graph siano complete.

Vantaggi e svantaggi del RDF Graph [7]

I principali vantaggi dei RDF Graph sono:

- *Standardizzazione*: tutti i grafi della conoscenza basati su RDF utilizzano la stessa struttura standard e la stessa semantica formale per archiviare e rappresentare i dati insieme ad un linguaggio di interrogazione standard.
- *Interoperabilità*: RDF Triple Stores segue uno standard supportato dal W3C che permette a grafi basati su RDF di integrare e scambiare informazioni.
- *Estensibilità*: i RDF Graphs consentono agli utenti di aggiungere nuovi nodi e relazioni senza richiedere la ricostruzione del database.

Alcuni svantaggi dei RDF Graph sono:

- *Complessità della ricerca*: l'esecuzione di una ricerca approfondita in RDF Graphs di grandi dimensioni è un'attività complessa in quanto richiede l'attraversamento di ogni relazione.
- *Rigorosa aderenza agli standard*: tutte le informazioni memorizzate in RDF devono essere sotto forma di triple.

2.2.2 Property Graph

Un **Property Graph** è un grafo diretto etichettato in cui nodi e archi possono avere un insieme di coppie proprietà-valore (figura 5). Un nodo rappresenta un'entità, un arco rappresenta una relazione tra entità, una proprietà rappresenta una caratteristica specifica di entità o relazione.

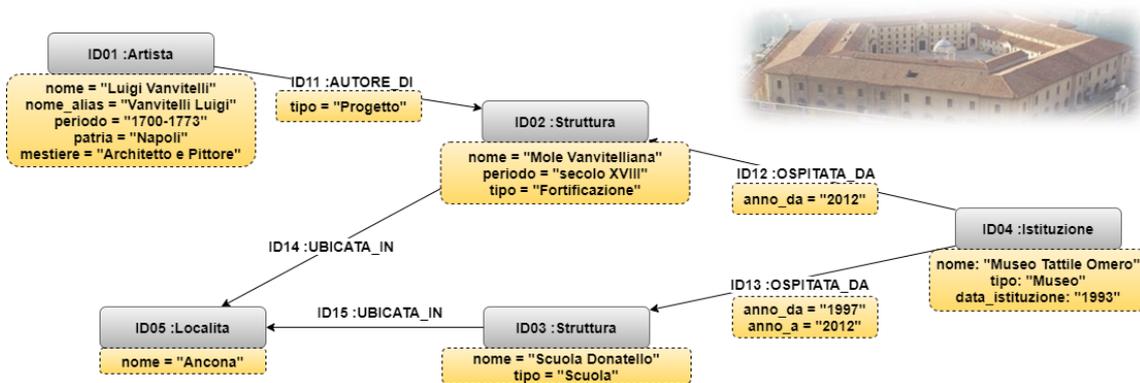


Figura 5 – Esempio di Property Graph con l'entità che rappresenta la Mole Vanvitelliana

Definizione [5]

Siano dati L insieme infinito di etichette (per nodi e archi), P insieme infinito di nomi di proprietà, U insieme infinito di valori atomici, T insieme finito di tipi di dati, SET+(X) l'insieme di tutti i sottoinsiemi finiti di X, escluso l'insieme vuoto.

Un **Property Graph** è una tupla $G = (N, E, \rho, \lambda, \sigma)$ dove:

1. N è un insieme finito di nodi;
2. E è un insieme finito di archi tale che E non ha elementi in comune con N;
3. $\rho : E \rightarrow (N \times N)$ è una funzione che associa ogni arco con una coppia di nodi;
4. $\lambda : (N \cup E) \rightarrow SET+(L)$ è una funzione che associa un nodo/arco con un insieme di etichette da L;
5. $\sigma : (N \cup E) \times P \rightarrow SET+(U)$ è una funzione parziale che associa nodi/archi a proprietà, e per ciascuna proprietà assegna un insieme di valori da U.

In figura 5 l'insieme N contiene 5 elementi (ID01, ID02, etc.), l'insieme E ne contiene 5 (ID11, ID12, ecc.) e alcune mappature di esempio sono:

```

    rho(ID11)=(ID01, ID02)
    lambda(ID01)=Artista
    lambda(ID11)=AUTORE_DI
    sigma(ID11)={(tipo, "Progetto")}
    sigma(ID02)={(nome, "Mole Vanvitelliana"), (periodo, "secolo XVIII"), (tipo, "Fortificazione")}
  
```

I Property Graph [3] [6] [7] costituiscono un modello che offre molta flessibilità in sistemi complessi. Vengono anche detti *Labeled Property Graph* (LPG), dato che alle entità e alle loro relazioni si associano *label* (etichette) che le raggruppano e ne definiscono la categoria (concetto riconducibile a quello di classe). Le etichette vengono anche utilizzate per definire vincoli comuni e per aggiungere indici. Se le etichette utilizzate per modellare i dati e le relative connessioni sono pertinenti, le strutture e le relazioni risultano immediatamente comprensibili agli esseri umani e i dati possono essere facilmente interrogati.

Non esistendo uno standard, ogni implementazione [10] è leggermente diversa. In generale i nodi possono avere da zero a molte etichette di testo che ne rappresentano il tipo, gli archi hanno un tipo e se sono diretti possono essere esplorati e interrogati in entrambe le direzioni. I valori delle proprietà associate ai nodi e agli archi possono avere tipi di dati che dipendono dal fornitore.

Nei Property Graph DB i nodi e gli archi hanno un ID interno assegnato dal database e il riferimento viene eseguito tramite stringhe di testo. Non esiste una serializzazione standard e di solito i dati possono essere caricati da file in formato CSV.

Vantaggi e svantaggi del Property Graph [7]

I principali vantaggi dei Property Graph sono:

- *Semplicità*: i grafi sono semplici e veloci da configurare e utilizzare.
- *Navigabilità*: i grafi sono più facili da attraversare senza limitazioni.
- *Dettaglio di informazioni*: le proprietà associate offrono maggiori dettagli su entità di dati e relative relazioni senza dover creare nodi aggiuntivi. L'interpretazione delle informazioni è lasciata all'utente.

Alcuni svantaggi dei Property Graph sono:

- *Scarsa interoperabilità*: la mancanza di standardizzazione rende difficile la condivisione o lo scambio di dati con database diversi. Gli identificatori univoci sono locali e non hanno significato per nessun altro database.
- *Vendor lock-in*: le organizzazioni che utilizzano un Knowledge Graph basato su Property Graph non possono integrare le proprie informazioni su più strumenti o sistemi. La probabilità di essere vincolati ad un unico fornitore è elevata.

2.3 Differenze tra i modelli RDF e Labeled Property Graph

Giuseppe Futia nell'articolo *Knowledge Graph Perspectives: building bridges from RDF to LPG* [11] illustra la duplice prospettiva che caratterizza i Knowledge Graph:

- I. La prima prospettiva (modello di dati RDF) si concentra sulla rappresentazione della conoscenza. Il grafo è codificato come una raccolta di istruzioni formalizzate ed ha l'obiettivo di standardizzare la pubblicazione e la condivisione dei dati sul Web, garantendo l'interoperabilità semantica e favorendo il *reasoning* sulle affermazioni disponibili.
- II. La seconda prospettiva (implementata nel modello *Labeled Property Graph*) si concentra sulla struttura del grafo. Questa visione sottolinea le caratteristiche dei dati, consente l'analisi e la visualizzazione, e favorisce lo sviluppo di sistemi di *machine learning* basati sui grafi per dedurre ulteriori informazioni.

I modelli RDF Triple Store e Labeled Property Graph sono entrambi adatti per l'archiviazione e la gestione dei Knowledge Graphs, ma differiscono in aspetti che si possono riassumere schematicamente [7]:

Aspetto	RDF Triple Stores	Labeled Property Graphs
<i>Rappresentazione</i>	Entità e relazioni sono rappresentate con la struttura <i>soggetto-predicato-oggetto</i>	Entità e relazioni hanno attributi associati rappresentati in coppie <i>chiave-valore</i>
<i>Linguaggi di interrogazione</i>	Linguaggio standard SPARQL	Ogni implementazione può avere un proprio linguaggio
<i>Struttura interna</i>	Entità e relazioni non hanno una struttura interna, l'identità è definita dagli URI	Entità e relazioni hanno una struttura interna, l'identità è determinata dall'insieme di etichette e proprietà
<i>Focus</i>	Standardizzazione e interoperabilità	Archiviazione dei dati e velocità delle query
<i>Casi d'uso</i>	Utile con dataset che cambiano lentamente. Perfetto per scenari che richiedono <i>reasoning</i> /inferenza o dove sono richieste informazioni da altri archivi	Utile con Knowledge Graph di grandi dimensioni, con dataset dinamici che richiedono frequentemente un attraversamento approfondito

Il modello di dati RDF fornisce una base più ricca e semanticamente coerente rispetto ai Property Graph [12]. Se questi ultimi sono tipicamente incentrati sui nodi, RDF Triple è incentrato sugli archi.

Il principale elemento di differenziazione [13] tra Property Graph e RDF Graph è la possibilità di avere attributi/relazioni sulle proprietà. Vantaggi del Web Semantic non presenti nei Property graphs sono:

1. *Semantica formale*. OWL è un linguaggio costruito su RDF ed ha una semantica formale che corrisponde ad un sottoinsieme decidibile della logica del primo ordine. I *reasoner* automatizzati possono convalidare un'ontologia per assicurarsi che non abbia contraddizioni e se l'ontologia risulta priva di contraddizioni si possono eseguire inferenze automatiche, come mantenere valori inversi e transitivi, ristrutturare la gerarchia di classi, riconoscere i tipi di un'istanza in base ai valori.
2. *Linked Data*. È possibile interrogare qualsiasi Knowledge Graph conforme al W3C pubblicato su Internet. Esistono centinaia di tali fonti con petabyte di dati utili, strutturati e gratuiti (es. DBpedia, HealthData.gov, Geonames).
3. *Vocabolari di settore*. Poiché OWL e RDF sono standard W3C, esistono centinaia di vocabolari verticali di settore riutilizzabili (es. SNOMED per l'assistenza sanitaria, FIBO per i servizi finanziari, FOAF per i social network).
4. *Vocabolari tecnici*. Esistono centinaia di vocabolari orizzontali per domini tecnici creati dal W3C e da altre organizzazioni, come Dublin Core per i metadati e SKOS per l'organizzazione e la categorizzazione della conoscenza.
5. *Semantic Web*. Oltre al linguaggio OWL che agisce ad un livello superiore rispetto a quello su cui lavorano RDF e Property graphs, esistono altre tecnologie, come SHACL (Shapes Constraint Language) per la definizione e l'applicazione dei vincoli di integrità dei dati e SWRL (Semantic Web Rule Language), che fornisce un motore di regole di “concatenamento in avanti” molto potente.

Altre differenze interessanti sono evidenziate nell'articolo *Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs* [9].

RDF è stato progettato come un framework per la pubblicazione e lo scambio di dati tra un ampio gruppo di stakeholder in un formato strutturato. LPG è stato sviluppato principalmente per l'utilizzo dei dati, con lo scopo di archivarli e interrogarli nel modo più efficiente possibile. A causa della mancanza di struttura interna dell'RDF, le proprietà dei nodi possono essere descritte solo aggiungendo nuovi nodi o letterali. Generalmente i diversi LPG suppongono l'ipotesi di mondo chiuso, mentre i concetti del linguaggio dello schema RDF si basano su un'ipotesi di mondo aperto, cosa che apre molte possibilità di inferenza e generalmente si adatta meglio al concetto di Semantic Web.

I grafi RDF e LPG differiscono notevolmente in termini di complessità. Il grafico RDF è più atomico mentre i Labeled Property Graph sono più compatti, dato che le proprietà delle entità sono memorizzate all'interno dei nodi o degli archi. Un vantaggio di avere proprietà sugli archi è che questo permette di creare facilmente relazioni temporali e relazioni ponderate, utilizzabili per algoritmi di analisi di rete, mentre i costrutti RDF temporali sono più ingombranti. RDF utilizza URI per definire nodi e relazioni, quindi non è possibile creare più istanze della stessa relazione.

3 K-MAG, il grafo dei beni artistici e culturali marchigiani

Come esempio concreto di applicazione si presenta K-MAG (Knowledge Marche Art Graph), un grafo di conoscenza costituito da informazioni relative ai beni culturali artistici e archeologici della regione storico-geografica marchigiana, per il quale è stato ideato e realizzato un logo mostrato in figura 6.

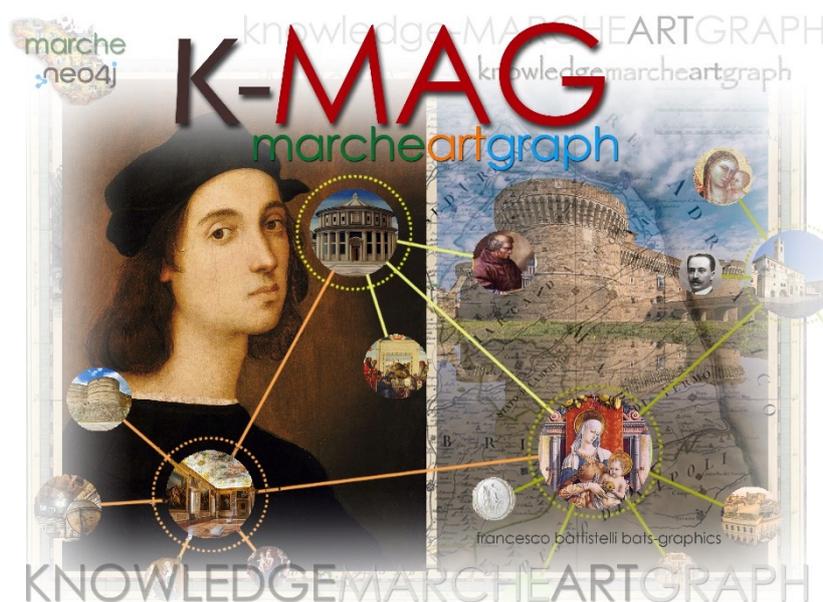


Figura 6 – Logo del Knowledge Graph K-MAG

3.1 Il dominio di interesse e il modello dei dati

Lo spazio fisico considerato è costituito dalle località che nel corso dei secoli hanno avuto una storia artistica e culturale comune, dalle corti medievali e rinascimentali sotto lo stato Pontificio fino all'età contemporanea sotto il Regno e la Repubblica Italiana. Il territorio corrisponde alle tre province storiche Ducato di Urbino, Marca Anconitana e Marca di Camerino.

Considerato che a livello comunale non c'è stata una rilevante discontinuità nella suddivisione amministrativa nel passaggio dallo Stato della Chiesa al Regno d'Italia [14], il dominio di interesse comprende le opere artistiche conservate nelle località appartenenti alle Delegazioni dello Stato Pontificio nel periodo successivo alla Restaurazione post-napoleonica (1816-1860) e alle Province marchigiane dall'Unità d'Italia (1861) a oggi. Sono gli attuali comuni della Regione Marche, i comuni del Montefeltro recentemente ceduti alla Regione Emilia-Romagna, la zona di Gubbio ceduta nel 1861 alla Regione Umbria e il comune di Ancarani temporaneamente unito ad Ascoli (figura 7).



Figura 7 – Delegazioni dello Stato Pontificio e Province attuali della Regione Marche

Il punto di partenza è stato individuato nella localizzazione di beni artistici collegati ai rispettivi autori. Per la creazione del Knowledge Graph si è scelto un insieme di dati omogenei, ovvero le Guide Rosse del Touring Club [15], i cui indici degli autori (prevalentemente architetti, scultori e pittori) e delle cose (edifici, monumenti, istituzioni, etc.) sono stati digitalizzati per ottenere un insieme iniziale di nodi da importare nel grafo. I collegamenti tra gli autori e i siti contenenti le opere sono stati verificati attentamente, per risolvere similitudini e ambiguità e per ottenere una base informativa di partenza qualitativamente accettabile.

In una prima fase non si è ritenuto utile inserire informazioni dettagliate sulle singole opere artistiche e archeologiche. In un secondo tempo, per rendere il Knowledge Graph più adeguato a descrivere il dominio di interesse, si è scelto di estrarre informazioni dalle Guide TCI, dalla Guida artistica Electa [16] e dai siti web dei principali musei della Regione Marche, per arricchire K-MAG con i dati di oltre quattrocento opere artistiche e archeologiche notevoli.

Il modello E-R che descrive il dominio di interesse è illustrato in figura 8. Le entità, provviste di attributi e di un identificatore univoco, modellano i concetti di *Artista*, *Opera artistica o archeologica*, *Struttura architettuale*, *Istituzione culturale*, *Località geografica* e *Unità Amministrativa*. Le relazioni sono tipizzate ed è possibile stabilire una direzione privilegiata. In generale le relazioni non hanno grado superiore a due, e alcune entità risultano associate a sé stesse.

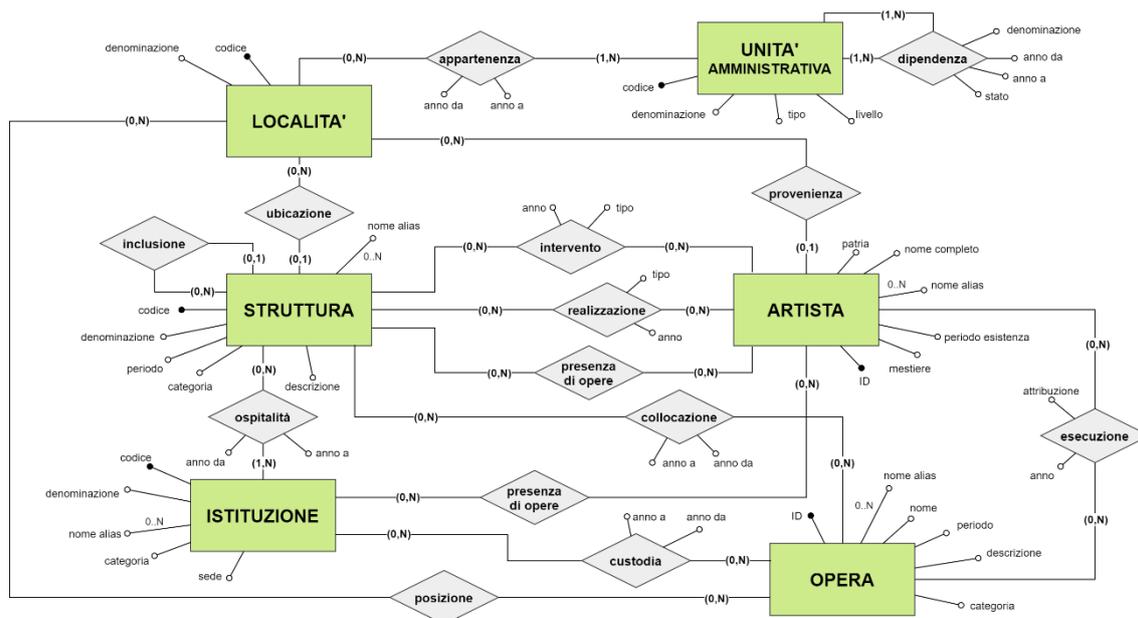


Figura 8 – Il modello Entity-Relationship di K-MAG

La scelta del Graph Model per rappresentare K-MAG in forma di grafo è ricaduta sul Labeled Property Graph, che permette di modellare le entità e le relazioni come nodi e archi diretti con specifiche *label* che ne descrivono la funzione (figura 9).

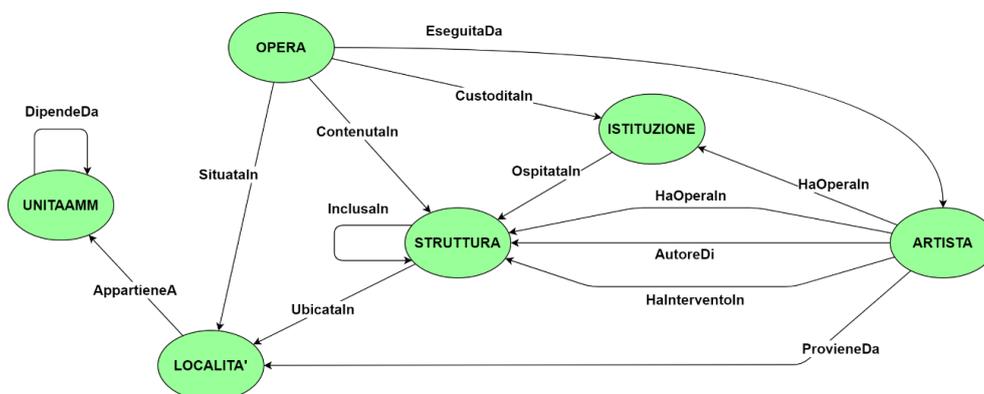


Figura 9 – Nodi e archi del grafo corrispondente al modello E-R

3.2 Implementazione di K-MAG in Neo4J

Come contenitore fisico del K-MAG è stato scelto Neo4j [17], l'implementazione più nota di Property Graph DB. Scritto in Java e disponibile gratuitamente, Neo4J è privo di schema, supporta le proprietà ACID e permette attraversamenti dei grafi ad alta velocità. I tipi di dati sono simili, ma non identici, ai tipi di dati del linguaggio Java e tramite Neo4j Browser è possibile interagire con il grafo e visualizzare i dati contenuti nel Graph DB. Neo4j mette a disposizione diverse tecniche che permettono di alimentare il grafo, di interrogarlo e di attraversarlo percorrendo nodi e relazioni.

Al termine delle operazioni di raccolta, bonifica e preparazione dei dati sono stati prodotti file in formato CSV contenenti le seguenti informazioni da caricare su Neo4j:

- **nodì Artista**
1053 artisti che sono nati o che hanno opere nella regione di interesse
- **nodì Istituzione**
205 istituti culturali (musei, biblioteche, ecc.) contenenti opere attribuite ad artisti
- **nodì Opera**
425 opere notevoli della regione di interesse, attribuite ad artisti o anonime
- **nodì Struttura**
1241 edifici, fortificazioni, monumenti, istituzioni, etc.) collegati agli Artisti precedenti
- **nodì Località**
359 località della regione di interesse notevoli e contenenti relative ai Siti precedenti
- **nodì UnitaAmministrativa**
322 unità amministrative dal livello comunale al livello provinciale
- **archi AppartieneA**
450 collegamenti tra Località e Comuni di appartenenza con periodi e denominazioni
- **archi DipendeDa**
813 collegamenti tra Unità Amministrative con i relativi periodi e denominazione del Comune
- **archi ProvieneDa**
373 collegamenti tra Artisti provenienti dalla regione di interesse e relative Località di nascita
- **archi UbicataIn**
1086 collegamenti tra Strutture e Località in cui sono ubicate
- **archi InclusaIn**
155 collegamenti tra Strutture contenute e Strutture contenitori
- **archi OspitataIn**
205 collegamenti tra le Istituzioni e le Strutture che le ospitano
- **archi AutoreDi**
329 collegamenti tra Artisti e Strutture realizzate o progettate
- **archi HaInterventoIn**
88 collegamenti tra Artisti e Strutture in cui hanno eseguito interventi parziali
- **archi HaOperaIn**
1184 collegamenti tra Artisti e Strutture in cui sono conservate le loro opere

- archi HaOperaIn
664 collegamenti tra Artisti e Istituzioni in cui sono conservate le loro opere
- archi ContenutaIn
177 collegamenti tra le Opere e le Strutture che le contengono
- archi CustoditaIn
234 collegamenti tra le Opere e le Istituzioni che le custodiscono
- archi SituataIn
18 collegamenti tra Opere esterne le Località in cui sono situate
- archi EseguitaDa
343 collegamenti tra le Opere e gli Artisti che le hanno eseguite

Dato che le entità fondamentali sono di sei classi, i nodi in Neo4J sono contraddistinti dal seguente insieme di etichette:

{ Autore, Istituzione, Località, Opera, Struttura, UnitAmm }

Gli attributi identificatori id sono proprietà con vincolo di unicità, mentre gli altri attributi corrispondono a proprietà opzionali. Si è sempre inserito almeno un nome comprensibile per poter interpretare i risultati delle interrogazioni. Ognuna delle relazioni individuate ha un nodo di partenza e un nodo di arrivo, un nome che la distingue e un insieme di proprietà opzionali.

Neo4J Browser permette di interrogare il DB, restituendo il risultato in diverse modalità. La figura 10 illustra come il browser visualizza un sotto-grafo risultato di una interrogazione che seleziona il nodo Mole Vanvitelliana e gli archi collegati. Le tipologie dei nodi sono differenziate con colori diversi, ed è possibile esplorare le relazioni di ogni nodo, consultare le proprietà di nodi e archi, modificare dimensioni e scala dell'area visibile, scaricare il risultato dell'interrogazione in diversi formati.

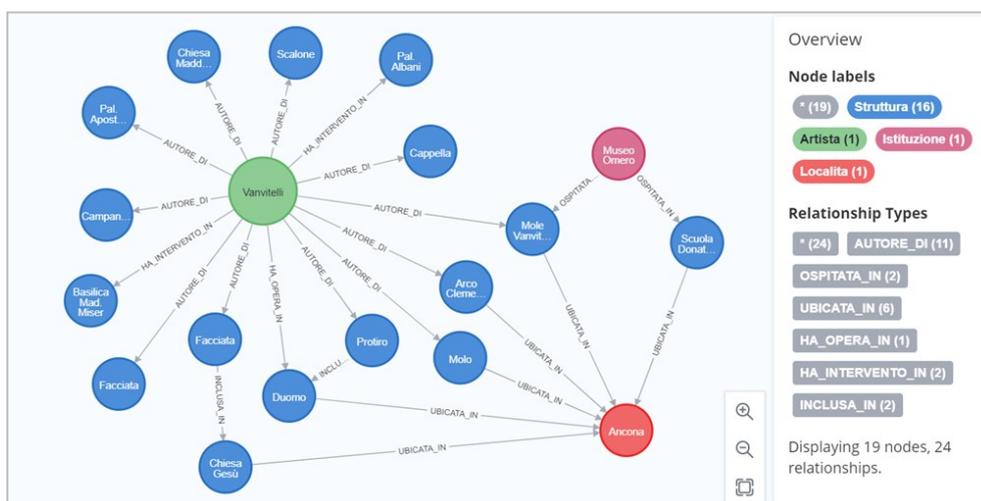


Figura 10 – Visualizzazione in Neo4j Browser del sotto-grafo relativo alla Mole Vanvitelliana

Al termine dell'importazione il grafo K-MAG contiene 3.605 nodi, collegati da 6.119 archi diretti. L'immagine del grafo completo è mostrata in figura 11, con 6 colori diversi associati alle *label* di nodo. Si evidenziano le parti maggiormente connesse tra loro e sulla sinistra si nota un cluster di nodi in colore giallo: sono beni culturali senza artisti collegati, custoditi al Museo Archeologico Nazionale delle Marche.

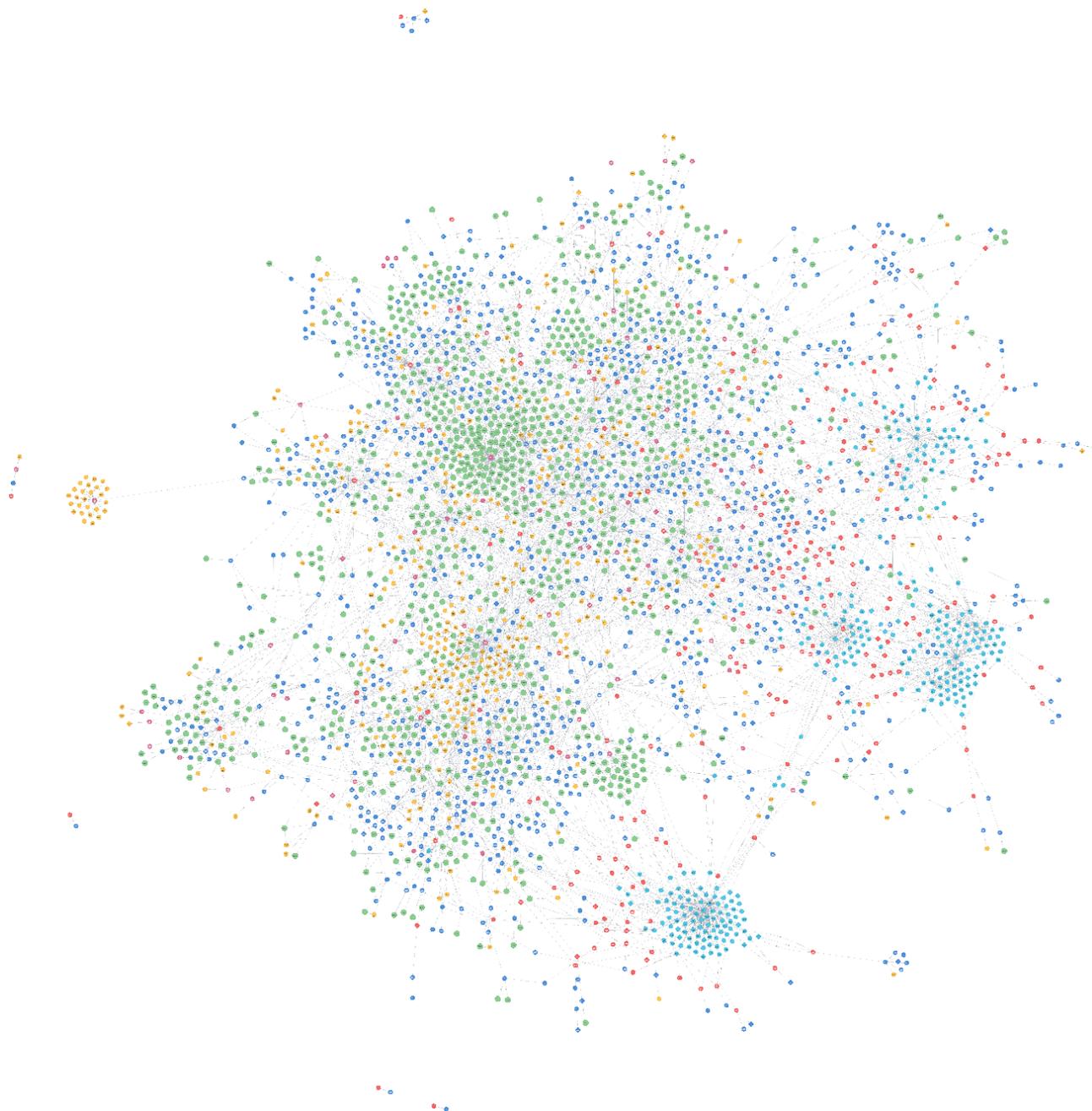


Figura 11 – Il grafo K-MAG completo

3.3 Incremento del contenuto informativo di K-MAG

Per incrementare il contenuto informativo, un'interessante fonte di informazione è il Catalogo dei Beni Culturali della Regione Marche [18], la cui banca dati (figura 12) contiene oltre 200.000 schede corredate di immagini e dati raccolti da studiosi e ricercatori in campagne di catalogazione del patrimonio culturale regionale. Un *wrapper* potrebbe estrarre il contenuto delle pagine HTML e salvarlo in dataset locali, per poter analizzare il testo e le descrizioni, confrontare risorse che trattano argomenti affini, riconoscere le stesse entità sui sistemi diversi, valutare somiglianze e differenze, integrare progressivamente le informazioni del Knowledge Graph. Il problema dell'uso di questo Catalogo è che non si conoscono le effettive strutture dei dati memorizzati presso la fonte.



Figura 12 – Catalogo dei Beni Culturali nel sito della Regione Marche

Un'imponente base di dati nazionale è il Catalogo Generale dei Beni Culturali [8], contenente oltre due milioni di schede relative a Beni storici e artistici sul territorio italiano, corredate di descrizioni testuali e immagini. Le schede del Catalogo (figura 13) sono consultabili nel sito per settore, per località o per altre tipologie di input, fino a raggiungere le informazioni desiderate.

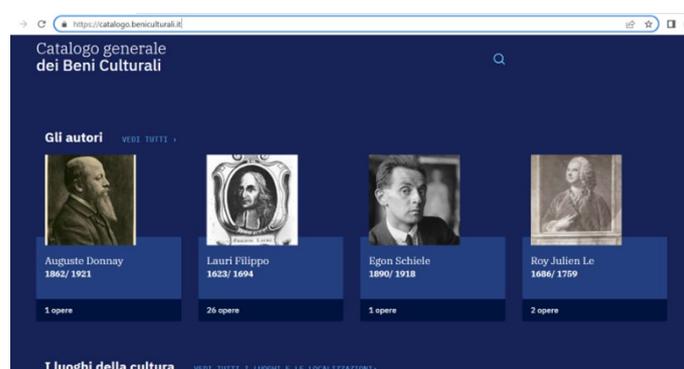


Figura 13 – Home page del Catalogo Generale dei Beni Culturali

4 Linguaggi di interrogazione

Per qualsiasi modello di dati è fondamentale avere a disposizione un linguaggio di query per interagire con i dati rappresentati dal modello. Per i grafi sono stati proposti numerosi linguaggi come SPARQL per interrogare RDF Graph e Cypher, Gremlin e G-CORE per interrogare i Property Graph.

SPARQL è il linguaggio di interrogazione dichiarativo sviluppato dal World Wide Web Consortium (W3C) come standard per l'interrogazione di dati RDF, particolarmente adatto per l'interrogazione di Linked Data e Semantic Web. Non esiste un linguaggio di query standard del settore per i Property Graph. Il principale linguaggio utilizzato nei Property Graph è Cypher, linguaggio dichiarativo sviluppato da Neo4j che imita la rappresentazione visuale di un grafo, con una sintassi che ricorda il linguaggio naturale.

4.1 Graph Pattern

Alla base dei linguaggi di interrogazione si possono delineare alcune primitive comuni e per ogni linguaggio di query strutturato per interrogare grafi esistono **Graph Pattern** che seguono lo stesso modello del Data Graph (figura 16). Il Graph pattern genera mappature dalle variabili¹ alle costanti del grafo, in modo tale che sostituendo le variabili con le costanti si ottiene un'immagine della mappatura inclusa nel corrispondente Data Graph.

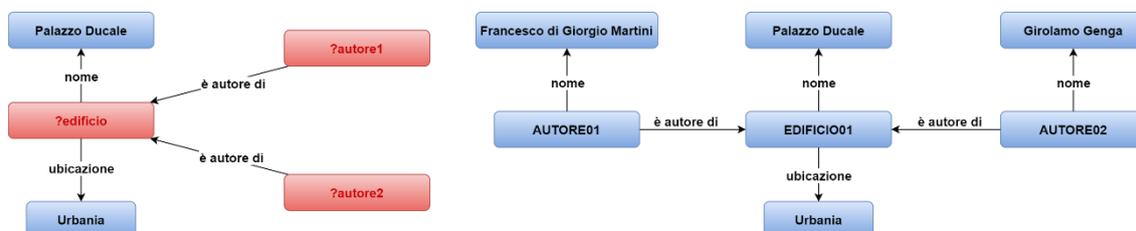


Figura 16 – Esempio di Graph pattern da mappare sul Data Graph di destra

¹ per distinguere le variabili alcuni linguaggi utilizzano come prefisso il punto interrogativo.

Definiamo il Basic Graph Pattern per il modello *Directed edge-labeled graph*², introducendo un insieme numerabile infinito di termini costituito da insiemi disgiunti di variabili e costanti.

Definizione [3]

Un **Basic Directed edge-labeled Graph pattern** è una tupla $Q = (N, E, L)$, dove N è un insieme di termini dei nodi, L è un insieme di termini degli archi ed $E \subseteq N \times L \times N$ è un insieme di archi.

In figura 16 N contiene costanti {"Palazzo Ducale", "Urbania"} e variabili {?edificio, ?autore1, ?autore2}, L contiene le costanti {"nome", "ubicazione", "autore di"}.

In alcune mappature più variabili vengono mappate sullo stesso termine, il che non sempre è desiderabile. Sono state proposte diverse semantiche da cui dipendono i risultati finali della valutazione di un Basic Graph Pattern:

- una semantica basata sull'**omomorfismo**, che consente di mappare più variabili sullo stesso termine; se Q è il Basic Graph pattern e G è il Data Graph, i risultati con questa semantica sono definiti come l'insieme delle mappature $Q(G)$.
- una semantica basata sull'**isomorfismo**, che richiede che le variabili su nodi e archi siano mappate su termini univoci; in questo caso mappature che inviano due variabili di arco alla stessa costante e le mappature che inviano due variabili di nodo alla stessa costante possono essere escluse dai risultati.

Linguaggi diversi possono adottare semantiche diverse per valutare Graph pattern: ad esempio SPARQL adotta una semantica basata sull'omomorfismo, Cypher adotta sugli archi una semantica basata sull'isomorfismo sugli archi, ma consente la mappatura di più variabili sullo stesso nodo (figura 17).

<i>semantica basata su omomorfismo (SPARQL)</i>		<i>semantica di Cypher</i>	
"autore1"	"autore2"	"autore1"	"autore2"
"Girolamo Genga"	"Francesco di Giorgio Martini"	"Girolamo Genga"	"Francesco di Giorgio Martini"
"Francesco di Giorgio Martini"	"Girolamo Genga"	"Francesco di Giorgio Martini"	"Girolamo Genga"
"Francesco di Giorgio Martini"	"Francesco di Giorgio Martini"	"Francesco di Giorgio Martini"	"Girolamo Genga"
"Girolamo Genga"	"Girolamo Genga"	"Girolamo Genga"	"Girolamo Genga"

Figura 17 – Risultati della valutazione del pattern in figura 16 con le diverse semantiche

² Per le altre tipologie di grafo, il Basic Graph Pattern viene definito in modo analogo: per il Property Graph si introducono insiemi di termini proprietà-valore, consentendo alle variabili di sostituire le costanti in ogni posizione.

Linguaggi di interrogazione come SPARQL e Cypher consentono il pieno utilizzo di operatori relazionali, dando origine a Complex Graph pattern costruiti a partire dai Basic Graph pattern mediante operatori di proiezione, selezione, unione, join, etc. Sulla base di questi operatori si possono definire altri operatori sintattici aggiuntivi, che non aggiungono espressività (es. left-join).

Definizione [3]

Un **Complex graph pattern** (C.G.P.) è definito ricorsivamente:

- Se Q è un basic graph pattern di base, allora Q è un C.G.P.
- Se Q è un CGP e $V \subseteq \text{Var}(Q)$, allora $\pi_{V(Q)}$ è un C.G.P..
- Se Q è un CGP e R è una condizione di selezione con connettivi booleani e di uguaglianza ($\wedge, \vee, \neg, =$), allora $\sigma_{R(Q)}$ è un C.G.P.
- Se sia Q_1 che Q_2 sono C.G.P., allora sono C.G.P. anche $Q_1 \bowtie Q_2$, $Q_1 \cup Q_2$, $Q_1 - Q_2$ e $Q_1 \triangleright Q_2$.

Graph pattern complessi possono dare origine a risultati duplicati, e anche in questo caso i diversi linguaggi di query offrono due semantiche: la semantica *bag* preserva i duplicati in base alla molteplicità delle mappature sottostanti, mentre la semantica *set* (tipicamente invocata con una parola chiave DISTINCT) rimuove i duplicati dai risultati.

Linguaggi specifici per i grafi possono supportare altre funzionalità, come l'aggregazione (GROUP BY, COUNT, ecc.), filtri più complessi, query su intervalli e supporti per l'implicazione. Una caratteristica chiave che distingue i linguaggi di query su grafo sono i **Navigational Graph patterns**, ovvero la capacità di includere *path expressions* nelle query. Una *path expression* r è un'espressione regolare che consente di associare percorsi di lunghezza arbitraria tra due nodi. In questo caso è possibile trovare tutte le coppie di artisti che hanno un percorso comune da autore che li collega nel grafo (es. la coppia (x_1, x_n) , dove x_1 è co-autore di x_2 per l'opera y_{12} , x_2 è co-autore di x_3 per l'opera y_{23} , e così via fino a x_n).

4.2 I linguaggi SPARQL e Cypher

SPARQL [7] [10] [19] [20], acronimo ricorsivo di *SPARQL Protocol And RDF Query Language*, è un linguaggio basato su Triple Store e recupera informazioni dai grafi in base alla corrispondenza con Graph pattern soggetto-predicato-oggetto.

Oltre ad essere un linguaggio di query completo supportato da RDF Store, è un protocollo che consente di inviare richieste di query agli endpoint tramite http. Le prestazioni variano a seconda del RDF Store utilizzato e della complessità della query. Il portale *Linked Open Data* dei Beni Culturali [12] contiene un end point SPARQL (figura 18) dal quale è possibile effettuare interrogazioni su RDF Store ottenendo risposte in formato tabellare HTML o in altri formati (es. CSV, JSON).

The screenshot shows a web browser window with the URL `dati.cultura.gov.it/sparql`. The page title is "Virtuoso SPARQL Query Editor". The "Query Text" field contains the following SPARQL query:

```
SELECT ?bc, ?label WHERE {
  ?bc rdfs:label ?label ;
  rdfs:type arco:HistoricOrArtisticProperty ;
  rdfs:label arco:RovellaCulturale@en@it ;
  arco:scatolidentificaz. "*" }
```

The "Results Format" is set to "HTML". The "Execution timeout" is 0 milliseconds. The "Options" section has "Strict checking of void variables" checked. The "Run Query" button is visible.

The results table has two columns: "bc" and "label". It lists various historical and artistic properties, such as "figura di soldato (monumento ai caduti) - ambito marchigiano (secondo quarto XXV)@en", "ex voto, serie - bottega marchigiana (sec. XIX)@en", "zoocollatura - bottega marchigiana (sec. XVIII)@en", "zoocollatura - bottega marchigiana (sec. XVIII)@it", "acquasantiera - bottega marchigiana (sec. XVIII)@en", "acquasantiera - bottega marchigiana (sec. XVIII)@it", "stallo, serie - bottega marchigiana (sec. XIX)@en", "stallo, serie - bottega marchigiana (sec. XIX)@it", "porta - bottega marchigiana (sec. XVIII)@en", "porta - bottega marchigiana (sec. XVIII)@it", "epigrafe funeraria di Caterina Agostini (lapide) - bottega marchigiana (sec. XIX)@en", "epigrafe funeraria di Caterina Agostini (lapide) - bottega marchigiana (sec. XIX)@it", "ancona by Baldi Lazzaro (sec. XVII)@en", "ancona di Baldi Lazzaro (sec. XVII)@it", "banco - bottega marchigiana (fine/inizio secc. XVIII- XIX)@en", "banco - bottega marchigiana (fine/inizio secc. XVIII- XIX)@it", "Madonna con Bambino (dipinto) - ambito marchigiano (sec. XIX)@en", "Madonna con Bambino (dipinto) - ambito marchigiano (sec. XIX)@it", "San Barnaba orante (dipinto) by Guerrieri Giovanni Francesco (primo quarto sec. XVII)@en", and "San Barnaba orante (dipinto) di Guerrieri Giovanni Francesco (primo quarto sec. XVII)@it".

Figura 18 – Richiesta su end point SPARQL del Ministero dei Beni Culturali e relativa risposta

Esempi di interrogazioni SPARQL sul portale *Linked Open Data* dei Beni Culturali

Una generica interrogazione SPARQL che restituisce tutti i beni culturali del Catalogo appartenenti alla classe "Bene storico o artistico" è la seguente:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX arco: <https://w3id.org/arco/ontology/arco/>
SELECT ?bc WHERE {?bc rdf:type arco:HistoricOrArtisticProperty }
```

L'istruzione PREFIX permette di creare prefissi utili per alleggerire il codice. Il Graph pattern è una singola tripla composta da un soggetto generico (la variabile ?bc), dal predicato `rdf:type` che indica la relazione di appartenenza e da un oggetto della classe `arco:HistoricOrArtisticProperty`.

Semplici *Basic Graph pattern* si realizzano con join di più triple o in notazione abbreviata per i Triple pattern riferiti allo stesso soggetto:

```
SELECT ?bc, ?label WHERE {
  ?bc dc:title "dipinto" .
  ?bc rdfs:label ?label .
  ?bc rdf:type arco:HistoricOrArtisticProperty }
```

```
SELECT ?bc, ?label WHERE {
  ?bc dc:type "dipinto" ;
  rdfs:label ?label ;
  rdf:type arco:HistoricOrArtisticProperty }
```

Vengono restituite le istanze della classe `arco:HistoricOrArtisticProperty` collegate con `dc:type` al letterale "dipinto" e i valori della proprietà `rdfs:label`.

Cypher [20] [21] è basato sul matching di Graph pattern, quindi è possibile costruire interrogazioni che individuano nodi che hanno relazioni particolari con altri nodi. Il Graph pattern rappresenta i nodi tra parentesi tonde (...), gli archi con il simbolo `-[...]-` e gli archi diretti con il simbolo `-[...]->`, con tre elementi opzionali: una variabile (che non può essere preceduta dal simbolo `?`), l'etichetta preceduta dai due punti, i valori delle proprietà inseriti tra parentesi graffe.

Neo4J Browser permette di interrogare il DB tramite script Cypher e restituisce il risultato in modalità grafo, tabella, testo o codice JSON (figura 19).

The screenshot displays the Neo4J Browser interface with a Cypher query and its results in three different views: graph, table, and JSON.

Query: `neo4j> MATCH (op:Opera) RETURN op ORDER BY op.nome`

Graph View: Shows a network of nodes and relationships. The nodes are labeled with their types and counts: **Artista (1)**, **Istituzione (8)**, **Struttura (5)**, **Località (5)**, **HA_OPERA_IN (8)**, **OSPITATA_IN (5)**, and **UBICATA_IN (6)**. It indicates "Displaying 18 nodes, 0 relationships".

Table View: Shows a list of 18 records for the query. Each record contains properties for the `op` variable, such as `periodo`, `categoria`, `nome`, and `id`.

JSON View: Shows the query result as a JSON array of objects. Each object has keys for `identity`, `labels`, and `properties`.

Figura 19 - Risultato di una query su Neo4J Browser in modalità grafo, tabella, testo e codice

Con apposite istruzioni Cypher è possibile creare, importare, modificare ed eliminare dati nel grafo, nonché generare nuove relazioni non esplicitamente inserite a partire da relazioni esistenti tra i nodi. Cypher permette di attraversare i grafi con interrogazioni complesse, molto onerose o impossibili all'interno di basi dati relazionali, e di richiamare algoritmi noti in teoria dei grafi, come il cammino minimo tra due nodi (figura 20).



Figura 20 – Visualizzazione del cammino minimo tra due nodi in Neo4J

Esempi di interrogazioni Cypher sul Graph DB Neo4J K-MAG

Una generica interrogazione Cypher che restituisce tutti i nodi con etichetta Opera presenti nel grafo K-MAG è la seguente:

```
MATCH (op:Opera)
RETURN op
```

Se interessano solamente le opere eseguite da Raffaello si deve specificare il valore della proprietà richiesta associata al nodo:

```
MATCH (op:Opera)-[:ESEGUITA_DA]-(:Artista {nome:'Raffaello'})
RETURN op
```

È possibile ottenere lo stesso risultato grazie ad una istruzione WHERE:

```
MATCH (op:Opera)-[:ESEGUITA_DA]-(:ar:Artista)
WHERE ar.nome = 'Raffaello'
RETURN op
```

Le query possono contenere più istruzioni MATCH concatenate:

```
MATCH (is:Istituzione)<-[:HA_OPERA_IN]-(:ar:Artista {nome:'Crivelli C.'})
MATCH (is {tipo:'Museo'})-[:OSPITATA_IN]->(st:Struttura)-[:UBICATA_IN]->(x)
RETURN ar, is, st, x
```

Partendo dai musei che custodiscono le opere di Raffaello si può richiedere il sotto-grafo contenente gli autori di opere custodite negli stessi istituti culturali. L'istruzione OPTIONAL MATCH richiede una corrispondenza opzionale, in modo da ottenere una risposta anche per i musei in cui la corrispondenza è negativa:

```
MATCH (ar:Artista {id:'RAFFAELLO'})-[:HA_OPERA_IN]->(is:Istituzione)
OPTIONAL MATCH (is)<-[:HA_OPERA_IN]->(x)
RETURN is, x;
```

È possibile trovare i nodi che hanno un legame con uno specifico nodo entro un numero predefinito di relazioni. Il seguente pattern permette di ottenere la lista degli artisti che soddisfano il requisito di essere collegati a Raffaello con un numero di archi compreso tra il minimo 1 e il massimo 3:

```
MATCH (ar:Artista {id:'RAFFAELLO'})-[*1..3]->(x:Artista)
RETURN DISTINCT x.nome
```

Se si vuole creare una nuova relazione che lega due nodi Artista distinti se questi hanno realizzato una stessa Struttura, si può usare la seguente istruzione che aggiorna il grafo (l'uso di MERGE al posto di CREATE evita di duplicare relazioni già presenti):

```
MATCH (a1)-[r1:AUTORE_DI]->(:Struttura)<-[r2:AUTORE_DI]->(a2)
WHERE a1 <> a2
MERGE (a1)-[r3:COAUTORE_DI]->(a2)
RETURN a1, r3, a2;
```

La seguente interrogazione richiede il percorso minimo in K-MAG tra lo scultore Pericle Fazzini e l'illustratore Fortunato Depero, restituito da Noe4J in figura 20:

```
MATCH p = shortestPath(({nome:'Fazzini'})-[*]->({nome:'Depero'}))
RETURN p
```

4.3 Interrogazioni Cypher e SPARQL a confronto

Per un confronto sintattico tra Cypher e SPARQL con esempi specifici sul dominio di interesse, si richiedono le seguenti informazioni sul DB Neo4J K-MAG e sull'endpoint SPARQL dei Beni Culturali:

1. elenco dei Musei della provincia di Macerata con le località in cui si trovano e il numero di artisti che hanno opere custodite al loro interno in ordine decrescente;
2. elenco dei dipinti custoditi nella Galleria Nazionale delle Marche, con il titolo, la data di esecuzione e l'autore attribuito, aventi Francesco come soggetto.

Le figure 21 e 22 descrivono i sotto-grafi nelle modellazioni RDF e LPG, contenenti le informazioni utili per costruire le interrogazioni precedenti con i Graph pattern necessari su RDF Store dei Beni Culturali e sul Property Graph K-MAG.

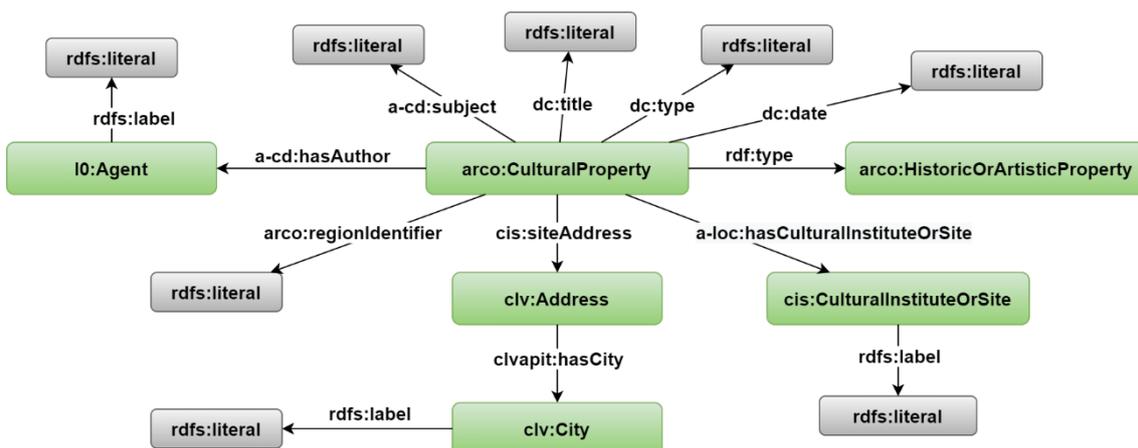


Figura 21 – RDF Graph dei Beni Culturali con le risorse utili per rispondere alle query

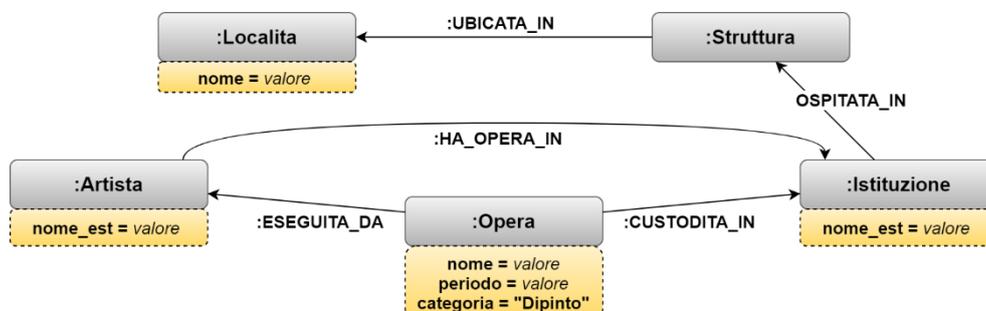


Figura 22 – Property Graph K-MAG con nodi, archi e proprietà utili per rispondere alle query

Le figure da 23 a 26 riportano le interrogazioni in codice SPARQL e Cypher, con le relative risposte in formato tabellare³. Le corrispondenze sintattiche sono evidenziate con lo stesso colore:

- le clausole **MATCH** (Cypher) e **WHERE** (SPARQL) confrontano i Graph pattern e associano i risultati alle variabili;
- le clausole **WHERE** (Cypher) e **FILTER** (SPARQL) specificano le condizioni, selezionando solamente i risultati che le soddisfano;
- le clausole **RETURN** (Cypher) e **SELECT** (SPARQL) eseguono una proiezione, specificando i valori legati alle variabili da restituire.

Codice Cypher

```

MATCH (ar:Artista)-[HA_OPERA_IN]->(is:Istituzione)
MATCH (is)-[OSPITATA_IN]->()-[UBICATA_IN]->(lc)-[APPARTIENE_A]->()-[d:DIPENDE_DA]->(ua)
WHERE ua.id='MC'
RETURN is.nome_est AS museo, lc.nome AS luogo, COUNT(ar) AS artisti
ORDER BY COUNT(ar) DESC

```

"museo"	"luogo"	"artisti"
"Pinacoteca Comunale"	"Porto Recanati"	48
"Pinacoteca comunale"	"Macerata"	18
"Pinacoteca-Museo Civico"	"Camerino"	15
"Museo Diocesano"	"Camerino"	14
"Pinacoteca comunale M. Moretti"	"Civitanova Alta"	12
"Museo Diocesano"	"Recanati"	9
"Pinacoteca comunale P. Tacchi-Venturi"	"San Severino Marche"	9
"Pinacoteca palazzo Ricci"	"Macerata"	9
"Museo Piersanti"	"Matelica"	7
"Pinacoteca"	"Corridonia"	7

Figura 23 – Query Cypher per ottenere l'elenco dei Musei

³ Le differenze quantitative nei risultati dipendono dal numero di informazioni registrate, più elevato nel Catalogo dei Beni Culturali rispetto al K-MAG.

Codice SPARQL

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX arco: <https://w3id.org/arco/ontology/arco/>
PREFIX a-cd: <https://w3id.org/arco/ontology/context-description/>
PREFIX a-loc: <https://w3id.org/arco/ontology/location/>
SELECT DISTINCT ?museo, ?luogo, COUNT(DISTINCT ?autore) AS ?artisti
WHERE { ?bc rdf:type arco:HistoricOrArtisticProperty ;
        arco:regionIdentifier ?regione ;
        a-cd:hasAuthor ?autore ;
        a-loc:hasCulturalPropertyAddress ?address .
        a-loc:hasCulturalInstituteOrSite/rdfs:label ?museo .
        ?address clvapit:hasProvince/rdfs:label ?provincia ;
        clvapit:hasCity/rdfs:label ?luogo .
FILTER (?provincia = 'MC')
} ORDER BY DESC(?artisti)

```

museo	luogo	artisti
"Pinacoteca e Musei Civici"	"Macerata"	416
"Pinacoteca Comunale - Galleria d'Arte Moderna Marco Moretti"	"Civitanova Marche"	113
"Pinacoteca Comunale "A. Moroni"'"	"Porto Recanati"	72
"Pinacoteca e Biblioteca Comunale Mozzi-Borgetti"	"Macerata"	41
"Pinacoteca Civica e Museo Archeologico"	"Camerino"	17
"Museo di S. Agostino"	"Visso"	12
"Pinacoteca Comunale"	"Treia"	11
"Museo delle Memorie Patrie Pollentine"	"Pollenza"	11
"Pinacoteca comunale P. Tacchi - Venturi"	"San Severino Marche"	11
"Museo Civico-Pinacoteca "Scipione Gentili"'"	"San Ginesio"	7
"Biblioteca Comunale "Mozzi - Borgetti"'"	"Macerata"	6
"Museo Comunale"	"Potenza Picena"	2
"Pinacoteca Civica "Benedetto Biancolini"'"	"Potenza Picena"	2
"Accademia Georgica"	"Treia"	2
"Museo "F. Corridoni"'"	"Corridonia"	1

Figura 24 – Query SPARQL per ottenere l'elenco dei Musei

Codice Cypher

```

MATCH (ar:Artista)-[:ESEGUITA_DA]-(op:Opera {categoria:
'Dipinto'})-[:CUSTODITA_IN]->(is:Istituzione)
WHERE op.nome CONTAINS 'Francesco' AND is.nome_est CONTAINS 'Galleria Nazionale
delle Marche'
RETURN op.nome AS Opera, op.periodo AS Periodo, ar.nome_est AS Autore

```

"Opera"	"Periodo"	"Autore"
"Stimmate di S. Francesco"	null	"Federico Fiori, detto Barocci"
"Incontro di S. Domenico e S. Francesco"	"secolo XVI"	"Girolamo Genga"
"Madonna col Bambino e Ss. Francesco e Giovanni Battista"	"secolo XVII"	"Simone De Magistris"

Figura 25 – Query Cypher per ottenere l'elenco dei dipinti

Codice SPARQL

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX arco: <https://w3id.org/arco/ontology/arco/>
PREFIX a-cd: <https://w3id.org/arco/ontology/context-description/>
PREFIX a-loc: <https://w3id.org/arco/ontology/location/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?soggetto, ?periodo, ?autore WHERE {
  ?bc dc:type "dipinto"
  ; rdf:type arco:HistoricOrArtisticProperty
  ; dc:title ?titolo
  ; a-cd:subject ?soggetto
  ; a-cd:hasAuthor/rdfs:label ?autore
  ; a-loc:hasCulturalInstituteOrSite ?cis .
  ?cis rdfs:label ?lcis
  OPTIONAL { ?bc dc:date ?periodo }
  FILTER (CONTAINS(lcase(?lcis),"galleria nazionale delle marche"))
  FILTER (REGEX(STR(?soggetto), "Francesco", "i")) }

```

soggetto	periodo	autore
"San Francesco d'Assisi"	"post 1553-ante 1597"	"Frangipane Niccolò (bottega)"
"San Francesco d'Assisi"	"1441-1460"	"Bartolomeo Di Tommaso (attribuito)"
"Francesco Maria II Della Rovere"	"1601-1625"	"Fiori Federico Detto Barocci (scuola)"
"Madonna con Bambino in gloria con Sant'Agostino, San Francesco, Sant'Elisabetta e San Giovannino"	"1590-1610"	"Cimatori Antonio - 1550 ca./ 1623"
"Madonna con Bambino in gloria con Sant'Agostino, San Francesco, Sant'Elisabetta e San Giovannino"	"1590-1610"	"Cimatori Antonio (1550 Ca./ 1623)"
"San Francesco d'Assisi e San Giacomo della Marca"	"1626-ante 1631"	"Lilli Andrea - 1570 ca./ post 1631"
"San Francesco d'Assisi e San Giacomo della Marca"	"1626-ante 1631"	"Lilli Andrea (1570 Ca./ Post 1631)"
"San Francesco d'Assisi con i Santi Bonaventura, Antonio da Padova e il Beato Leone"	"1626-ante 1631"	"Lilli Andrea - 1570 ca./ post 1631"

Figura 26 – Query SPARQL per ottenere l'elenco dei dipinti

4.4 Tradurre da SPARQL a Cypher: lo stato dell'arte

Le differenze tra i modelli RDF e Property Graph si riflettono sui linguaggi di interrogazione dichiarativi SPARQL e Cypher. Secondo Futia [11] le visioni racchiuse dalle due prospettive sui Knowledge Graph (rappresentazione della conoscenza in RDF, struttura del grafo nei LPG) hanno un impatto diretto sull'esperienza dell'utente. Nel caso dei grafi RDF, diversi tipi di utenti (sviluppatori, analisti di dati) interagiscono con i dati serializzati come istruzioni, e quando si utilizza SPARQL qualsiasi operazione eseguita sul grafo implica la manipolazione di tali istruzioni. In LPG, il linguaggio Cypher ha un'interazione più trasparente con la struttura del grafo in termini di nodi, archi e proprietà correlate. Questa differenza ha un impatto significativo sull'esperienza utente e sulle prestazioni.

La ridondanza dei dati è fortemente ridotta nel caso dei LPG perché tutte le proprietà dei dati diventano proprietà del nodo. Anche quando in RDF si hanno più istruzioni (uno per ogni proprietà del dato) per LPG c'è un'unica struttura dati, ovvero il nodo con le sue proprietà. La riduzione della ridondanza emerge anche confrontando interrogazioni equivalenti: la query Cypher è più compatta della query SPARQL, dato che SPARQL non è un motore di attraversamento di grafi e occorre specificare esplicitamente tutti i passi con un impatto cruciale sulle performance.

- Usando **SPARQL** lo sviluppatore definisce **COME** attraversare il grafico per recuperare le informazioni di cui ha bisogno e l'esecuzione della query dipende strettamente dal contenuto della query stessa.
- La logica alla base di **Cypher** è definire **COSA** fare, specificando quali dati recuperare in un modo simile a quello di umani che intendono navigare attraverso strutture a grafo, e le prestazioni dipendono dall'implementazione del motore di query.

Secondo Angles [22] la maggior parte dei linguaggi di query del Property Graph non ha una solida base formale. Questo solleva una sfida critica di interoperabilità delle query, poiché non è possibile definire una mappatura formale tra SPARQL e un linguaggio Property Graph, anche se diversi lavori per supportare l'interoperabilità utilizzano algebre relazionali per dimostrare le equivalenze tra le mappature. L'autore evidenzia la necessità di proporre un linguaggio di query standardizzato per Property Graph DB, che faciliti la definizione formale e lo studio dei metodi di trasformazione delle query.

Moreira e Ramalho [23] sostengono che tradurre SPARQL in Cypher è un processo complesso a causa delle differenze nella sintassi e nell'archiviazione dei dati. La completa correlazione è estremamente difficile e viene presentata l'estensione Neosemantics, che mira ad estendere le capacità di Neo4J con mappature tra ontologie e schemi del grafico.

Donkers et al. [9] sostengono che la crescente disponibilità di dati IoT richiede nuovi metodi di integrazione dei dati e confrontano due modelli di grafo. LPG supera il secondo in tutte le misure delle prestazioni, RDF è qualitativamente favorevole in alcuni scenari (es. ambienti multi-stakeholder, dove può condividere la conoscenza e fare inferenze). Gli autori propongono una futura ricerca per confrontare grandi dataset, sviluppando le query da semplici casi d'uso (qualità dell'ambiente interno, efficienza energetica, manutenzione) fino alle tipologie più complesse per testare gli schemi di trasformazione.

Nie et al. [24] propongono una nuova rappresentazione intermedia (GraphQ IR) per colmare il divario semantico tra il linguaggio naturale e i linguaggi di query sui grafi. GraphQ IR dimostra capacità di generalizzazione e robustezza superiori in contesti fuori distribuzione e con risorse limitate. Un limite sono le regole di composizione di GraphQ IR strettamente allineate con le frasi interrogative, formalismo non applicabile all'analisi semantica del dominio generale. Infine il parser semantico neurale lavora su un modello pre-addestrato con circa 139 milioni di parametri, non riproducibile senza risorse GPU adeguate.

Il lavoro recente più interessante è a cura di Zhao et al.: *S2CTrans: Building a Bridge from SPARQL to Cypher* [25], in cui si esplora la traduzione da SPARQL ai linguaggi di interrogazione Property Graph. Viene presentato il framework S2CTrans, che intende realizzare la traduzione utilizzando l'algebra relazionale dei grafi. Al momento il progetto realizza traduzioni da SPARQL a Cypher, presenta diverse limitazioni e gli autori prevedono di perfezionarlo ulteriormente anche per analizzare la traduzione da Cypher a SPARQL.

Dagli esempi visti e dagli studi consultati appare dunque non banale il problema di una traduzione diretta tra i due modelli valida ed applicabile per tutti i possibili domini di interesse.

5 Integrazione e analisi del grafo K-MAG

Per arricchire il Knowledge Graph originale con nuovi dati, analizzare i dati del grafo e confrontare il contenuto informativo con altre fonti, si sono utilizzate funzionalità e librerie del linguaggio Python, che permettono di interrogare direttamente i grafi con query Cypher su K-MAG e query SPARQL sull'end point del Ministero dei Beni Culturali. Le interrogazioni devono contenere clausole di condizione al cui interno possono esserci operazioni di confronto del contenuto informativo: in questo caso una opportuna scelta delle condizioni nel codice SPARQL è essenziale per ottimizzare le query. Nei processi di integrazione tra Knowledge Graph è importante riuscire a capire quando nomi diversi rappresentano la stessa entità e quando entità diverse hanno lo stesso nome. Il riferimento agli URI può risolvere questi problemi ma è sufficiente a risolvere ambiguità, attribuzioni controverse e classificazioni incerte.

Le funzioni di integrazione e analisi del Knowledge Graph sono state realizzate utilizzando le seguenti librerie del linguaggio Python:

- *numpy*, per l'analisi numerica con funzioni ad alto livello;
- *Pandas*, per la manipolazione e analisi dei dati in tabelle;
- *py2neo*⁴, per l'accesso ad un Graph DB Neo4J;
- *SPARQLWrapper*, per eseguire le query su un endpoint SPARQL;
- *networkX*, per la rappresentazione e l'analisi di reti complesse e di grafi;
- *nltk*, per l'elaborazione del linguaggio naturale;
- *matplotlib* e *seaborn*, per la generazione di grafici e strumenti di data visualization;
- *plotly* e *folium*, per visualizzare mappe interattive;
- *openCV*, per operazioni di Computer Vision.

⁴ Alla fine del mese di ottobre 2023 il progetto *py2neo* è stato dichiarato concluso dal suo creatore Nigel Small. È ancora possibile utilizzare la libreria, ma la manutenzione è terminata. Per sviluppi futuri è consigliabile utilizzare il driver ufficiale Neo4J o la libreria *neomodel*.

5.1 Entity Alignment e integrazione dei dati

In un Knowledge Graph la conoscenza del mondo è immagazzinata in entità che si riferiscono ad oggetti del mondo reale e relazioni che rappresentano connessioni tra quegli oggetti. Dato che è impossibile costruire un grafo che ricopra completamente un dominio reale, per migliorarne la completezza si integrano informazioni provenienti da altri grafi che contengono dati supplementari o complementari. Per combinare la conoscenza un passaggio cruciale è allineamento delle entità equivalenti, operazione nota come *Entity Alignment*.

Definizione [26]

Dati due Knowledge Graph $KG_1 = (N_1, E_1, L_1)$ e $KG_2 = (N_2, E_2, L_2)$, in cui N_i rappresenta l'insieme dei nodi entità, L_i l'insieme delle etichette delle relazioni, $E_i \subseteq N_i \times L_i \times N_i$ l'insieme degli archi diretti, l'*Entity Alignment* ha il compito di scoprire l'insieme delle coppie $S = \{(u, v) \mid u \in N_1, v \in N_2, u \leftrightarrow v\}$, dove \leftrightarrow è una relazione di equivalenza tale che u e v si riferiscono allo stesso oggetto del mondo reale.

La versione più semplice del problema (identificare in diverse fonti entità che si riferiscono alla stessa entità del mondo reale) è stata studiata da varie angolazioni [26].

Il processo di Entity Linking (EL) cerca di riconoscere entità in un testo scritto in linguaggio naturale e di collegarle ad entità corrispondenti in un Knowledge Graph (KG). L'Entity Resolution (ER), detta anche Entity Matching, Record Linkage o Deduplication, presuppone che siano presenti dati relazionali con attributi contenenti una grande quantità di informazioni testuali. ER utilizza funzioni di somiglianza o distanza per calcolare le somiglianze tra i valori degli attributi corrispondenti dei due oggetti. Questo processo consente metodi basati su regole o di *machine learning* per classificare coppie di oggetti come corrispondenti o non corrispondenti, in base ai punteggi di somiglianza calcolati.

Alcuni metodi di ER sono stati applicati ai modelli a grafo, dove emergono alcune peculiarità: le entità non hanno descrizioni testuali dettagliate, sono rappresentate dal loro nome e accompagnate da poche informazioni; a differenza dei database tradizionali, i KG si basano sulla Open World Assumption, in cui l'assenza di attributi per un'entità nel grafo non significa necessariamente che tali attributi non esistano nella realtà. I KG possono avere una semantica predefinita, come ontologia di assiomi logici o tassonomia di classi.

I metodi moderni di Entity Alignment (EA) affrontano il problema presupponendo che entità equivalenti in diversi KG condividano strutture locali simili e applicano tecniche di *machine learning* per misurare la distanza tra le entità in uno spazio di feature a bassa dimensionalità. Alcuni approcci puntano esclusivamente ad allineare le entità, altri si concentrano sull'allineamento delle relazioni, altri ancora mirano ad allineare le tassonomie delle classi. Esistono tecniche non supervisionate che operano direttamente sui dati di input senza training e tecniche che richiedono un apprendimento su mappature di entità allineate già conosciute; in entrambe si utilizzano le reti neurali e i recenti progressi delle tecniche di *deep learning*.

Per l'integrazione del K-MAG l'attenzione viene focalizzata sull'allineamento dei nodi del grafo, confrontando l'informazione contenuta all'interno di proprietà testuali e risorse *literal* collegate. La figura 27 mostra le parti di grafo che rappresentano la Pinacoteca di Ancona. Ogni entità nel proprio grafo ha identificatore univoco e le coppie di entità per cui vale la relazione di equivalenza sono evidenziate da linee rosse tratteggiate.

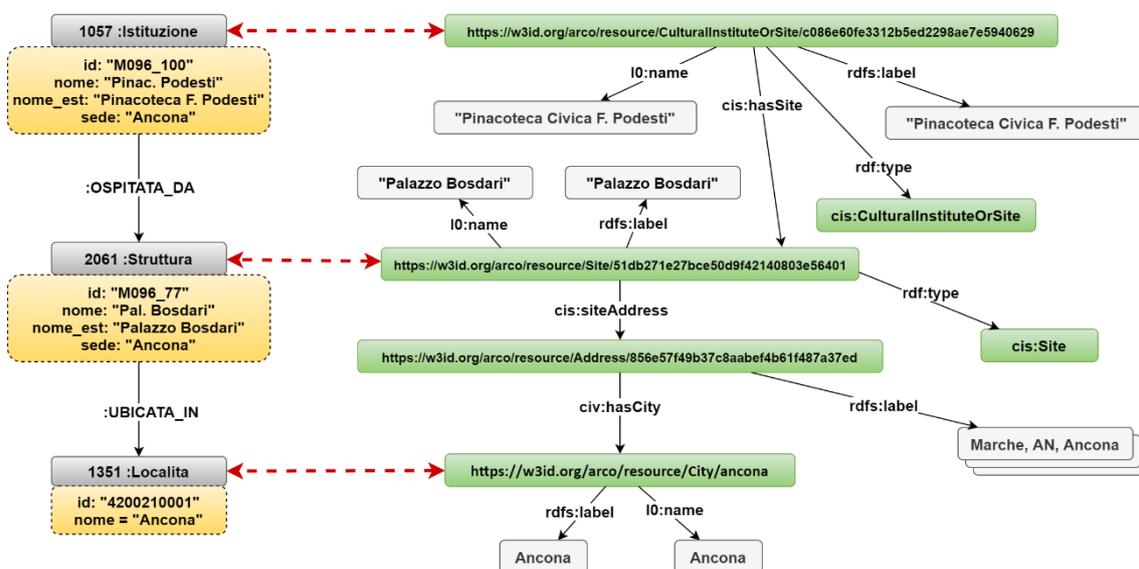


Figura 27 – Esempio di Entity Alignment tra due parti di Knowledge Graph.

L'attività di Entity Alignment è rivolta a trovare le coppie di entità equivalenti presenti nei due grafi, restituendo le URI delle entità corrispondenti alle entità presenti in K-MAG. Il confronto tra stringhe contenenti informazioni che accompagnano le entità da allineare può essere inizialmente effettuato applicando sulle stringhe gli operatori di uguaglianza e inclusione come sottostringa.

I risultati dei confronti relativi alla figura 27 sono mostrati in tabella 1. È evidente che in questo modo l'allineamento tra nodi K-MAG e risorse RDF non è particolarmente soddisfacente, visto che in questa modalità confronti che una persona giudicherebbe positivi non vengono riconosciuti come tali.

Entità	str1	str2	str1 = str2	str1 in str2
Museo	"Pinacoteca F. Podesti"	"Pinacoteca Civica F. Podesti"	<i>false</i>	<i>false</i>
Palazzo	"Palazzo Bosdari"	"Palazzo Bosdari"	<i>true</i>	<i>true</i>
Località	"Ancona"	"Marche, AN, Ancona"	<i>false</i>	<i>true</i>
Comune	"Ancona"	"Ancona"	<i>true</i>	<i>true</i>

Tabella 1 – Confronto tra stringhe per l'allineamento dei dati

Per migliorare l'allineamento dei dati si può introdurre il concetto di *similarità*, misura quantitativa della “vicinanza” tra due oggetti. Rappresentando le denominazioni e le descrizioni da confrontare come vettori, si utilizza la *cosine similarity*, definita⁵ come il coseno dell'angolo compreso:

$$\cos(x, y) = \frac{\langle x, y \rangle}{\|x\| \|y\|}$$

Per confrontare due nomi si determina il numero n dei termini contenuti in entrambe le frasi, scartando preventivamente eventuali termini non significativi, e si trasformano le due frasi da confrontare in vettori nello spazio dei termini R^n . I due vettori in ogni posizione hanno valore 1 se il termine corrispondente è contenuto nella frase e 0 in caso contrario. Calcolando il coseno dell'angolo compreso si ottiene una misura della distanza nell'intervallo $[0,1]$ delle frasi nello spazio R^n e definendo una opportuna soglia si può valutare se l'allineamento è positivo.

Nel caso precedente abbiamo considerato le frasi "Pinacoteca Civica F. Podesti" e "Pinacoteca F. Podesti" che non avevano superato i confronti di uguaglianza e di inclusione tra stringhe. Applicando la similarità, se nello spazio dei termini R^4 si sceglie arbitrariamente il vettore ['civica', 'f.', 'pinacoteca', 'podesti'], le frasi vengono trasformate in vettori con questa mappatura:

"Pinacoteca Civica F. Podesti" → (1,1,1,1)

"Pinacoteca F. Podesti" → (0,1,1,1)

Il calcolo della similarità come coseno dell'angolo compreso tra i due vettori fornisce il valore 0,866, che misura una buona vicinanza tra le frasi (1 corrisponde all'uguaglianza).

⁵ Nella formula $\langle x, y \rangle$ è un prodotto scalare tra due vettori e $\|x\|$ è la norma del vettore.

5.2 Implementazione del processo di integrazione K-MAG

I nodi del Knowledge Graph K-MAG sono partizionati in sei categorie tramite *label*, che si possono far corrispondere a specifiche classi definite all'interno delle ontologie ArCo⁶ pubblicate dal Ministero dei Beni Culturali. La tabella 2 riporta una mappatura tra i dati da allineare, con una misura quantitativa delle rispettive numerosità ricavate tramite opportune interrogazioni sull'end point SPARQL.

K-MAG			RDF Store Beni Culturali	
categoria	numerosità		classe in ontologie Arco e ontologie collegate	numerosità
:Artista	1.053	↔	l0:Agent (cyp:Person)	12.438
:Opera	425	↔	arco:HistoricOfArtisticProperty	135.836
			arco:ArcheologicalProperty	18.700
:Struttura	1.241	↔	arco:ArchitecturalOrLandscapeHeritage	25.828
			cis:Site	19.520
:Istituzione	205	↔	cis:CulturalInstituteOrSite	4.055
:Localita	359	↔	clv:City	399
			clv:AddressArea	1.134

Tabella 2 – Numerosità di entità nodo in K-MAG e risorse collegabili per classe in RDF Store

L'arricchimento del Knowledge Graph comporta sia l'aggiornamento dei dati già presenti, sia un incremento del numero di entità e relazioni presenti nel grafo. Per ottenere buoni risultati è importante acquisire informazioni sulle opere dal Catalogo Generale dei Beni Culturali, visto che la consultazione dei dati di un Bene Artistico, Archeologico e Architettonico implica la conoscenza sulle risorse collegate dalla catena di triple RDF.

⁶ ArCo utilizza una rete di ontologie per descrivere i diversi tipi di beni culturali e le schede catalografiche a essi associate; i prefissi e gli IRI dei moduli delle ontologie Arco e delle ontologie Cultural-ON e OntoPiA riusate sono:

```
arco:      https://w3id.org/arco/ontology/arco/
a-cat:    https://w3id.org/arco/ontology/catalogue/
a-dd:     https://w3id.org/arco/ontology/denotative-description/
a-loc:    https://w3id.org/arco/ontology/location/
a-cd:     https://w3id.org/arco/ontology/context-description/
cis:      http://dati.beniculturali.it/cis/
l0:       https://w3id.org/italia/onto/l0/
roapit:   https://w3id.org/italia/onto/RO/
tiapit:   https://w3id.org/italia/onto/TI/
clvapit:  https://w3id.org/italia/onto/CLV/
smapit:   https://w3id.org/italia/onto/SM/
language: https://w3id.org/italia/onto/Language/
```

In generale il workflow di integrazione del Knowledge Graph K-MAG con le informazioni presenti su RDF Store del Ministero dei Beni Culturali si può scomporre in tre macro-fasi evidenziate in figura 28. I tre blocchi in colore giallo racchiudono al loro interno le fondamentali attività critiche di Entity Alignment.

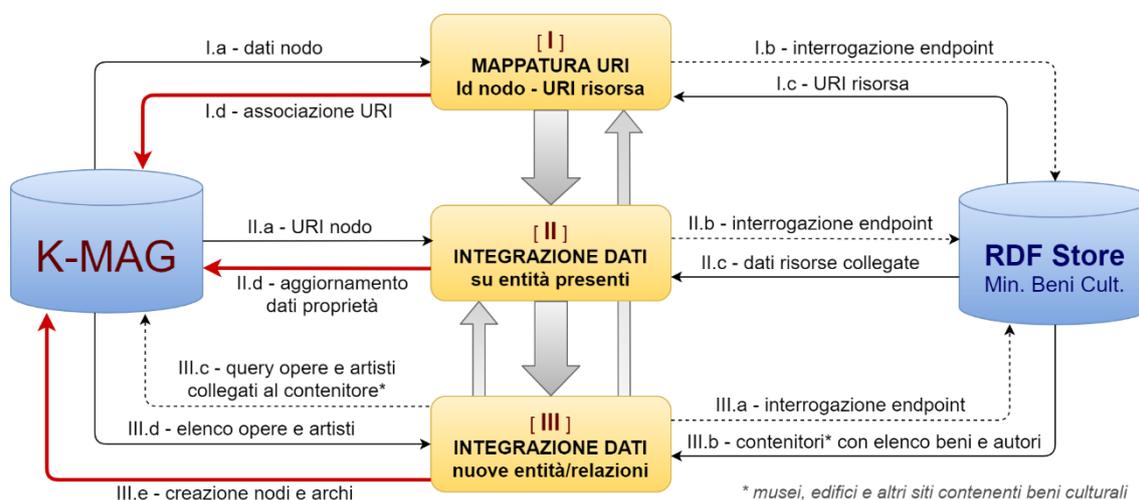


Figura 28 – Workflow del processo di integrazione K-MAG

- I. Viene eseguita una mappatura dei nodi presenti in K-MAG con le rispettive risorse URI, estraendo informazioni in base alle ontologie ArCo dei Beni Culturali.
- II. Vengono estratte le informazioni interessanti da RDF Store per gli Artisti, le Opere, le Strutture e le Istituzioni già presenti in K-MAG, per aggiornare o confermare i dati presenti e per integrare informazioni su proprietà non conosciute.
- III. Per incrementare il numero di nodi e degli archi presenti in K-MAG si interroga l'endpoint SPARQL del Ministero dei Beni Culturali per ottenere elenchi dei beni e degli autori collegati, iniziando dalle istituzioni culturali e proseguendo con edifici e altri siti artistici e culturali.

5.2.1 Macrofase I: mappatura dei nodi con URI dei Beni Culturali

Per l'allineamento tra nodi K-MAG e URI delle risorse occorre in primo luogo analizzare come vengono identificate le risorse RDF all'interno delle ontologie definite da ArCo o da altri soggetti.

La figura 29 mostra il sotto-grafo con le triple RDF da confrontare nei pattern matching delle interrogazioni SPARQL per ottenere le URI delle entità di interesse.

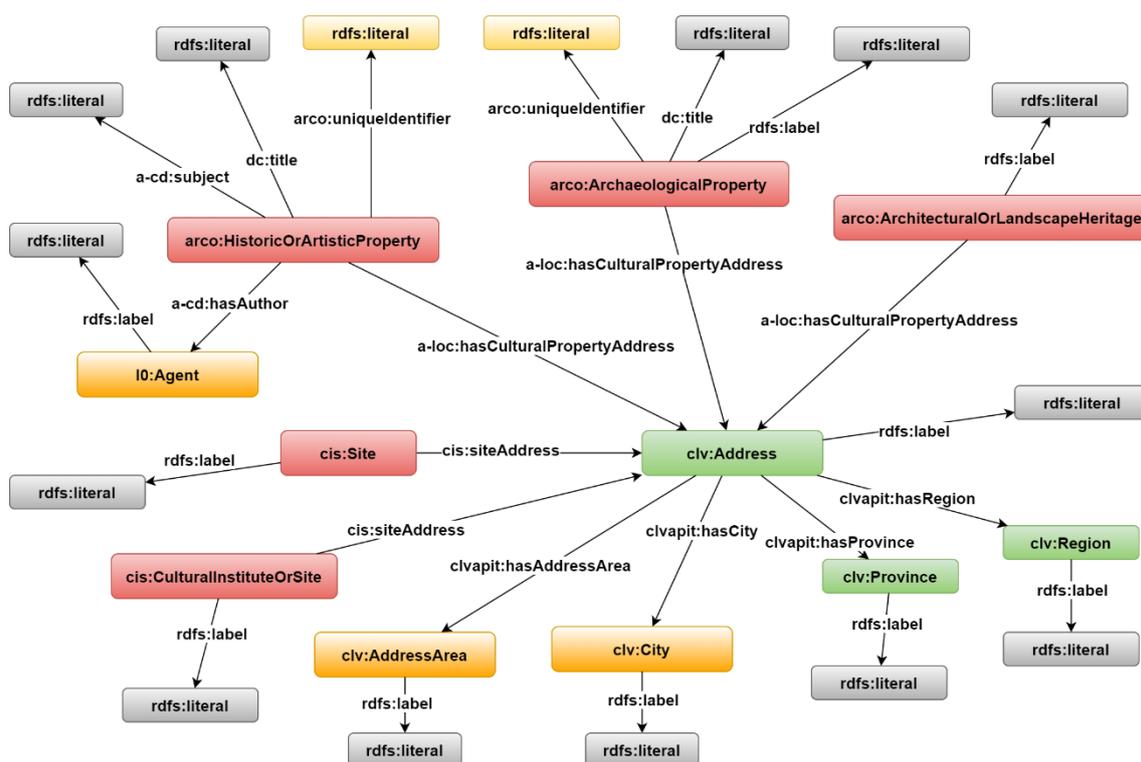


Figura 29 – RDF Graph dei Beni Culturali con le risorse utili per l'allineamento

In colore rosso sono indicate le Classi associate ad URI da interrogare direttamente su endpoint con buone prestazioni⁷, in arancio sono indicate le risorse associate ad URI ottenute percorrendo le triple. Le altre risorse rappresentate (istanze di classe in verde, *literal* in grigio o in giallo chiaro se hanno valori univoci) sono attraversate per navigare il grafo e per fare tutti i confronti necessari alla fase di allineamento.

Considerando il numero di risorse da confrontare e il numero di passaggi da fare per ottenere i dati, si è scelto ove possibile di operare in due sotto-fasi:

- si interroga l'endpoint SPARQL e si salva il risultato su file (algoritmo 1); la funzione TrovaUri riceve in input il percorso dell'endpoint e in base ai pattern matching e alla condizione di ricerca (specifici per ogni risorsa) produce in output un file CSV con l'elenco delle risorse che soddisfano le condizioni;
- i dati vengono allineati con i nodi e in caso di confronto positivo gli URI vengono aggiunti al Graph DB come valori di tipo lista (algoritmo 2); la funzione AbbinaUriNodo riceve in input la connessione al grafo e il file CSV e aggiorna il grafo aggiungendo ai nodi una proprietà contenente una lista delle URI allineate.

⁷ L'end point dei Beni Culturali per le risposta alle query SPARQL ha un timeout predefinito di 400 secondi e un limite di 30.000 record restituiti.

Algoritmo 1 – interrogazione dell'endpoint SPARQL

```

1: function TrovaUri(endpoint):
2:   sparql ← SPARQLWrapper(endpoint)
3:   sparql.setQuery(SELECT var WHERE {pattern_matching FILTER cond_Luogo})
4:   result ← sparql.query()
5:   df.columns ← result.variables
6:   for result in result.bindings:
7:     df ← result[variable].values
8:   df.to_csv(nomefile_uri)

```

Algoritmo 2 – allineamento tra nodi del grafo e URI delle risorse

```

1: function AbbinaUriNodo(graph_conn, nomefile_uri):
2:   graph ← Graph(graph_conn)
3:   list_nodi ← graph.run(MATCH pattern RETURN variables)
4:   df_uri ← read_csv(nomefile_uri)
5:   for nodo in list_nodi:
6:     for valore in nodo:
7:       for riga in df_uri:
8:         if valore.lower() in riga.lower():
9:           list_uri ← valore_uri
10:  list_abbinati ← [nodo,list_uri]
11:  for nodo_abbinato in list_abbinati:
12:    graph.run(MERGE nodo ON MATCH SET uri = list_uri)

```

Questa divisione in fasi non è conveniente per l'allineamento delle opere, perché il numero elevato dei Beni Culturali registrati nel Catalogo rende le query di selezione poco performanti. Considerando il numero ridotto delle opere presenti in K-MAG (425, di cui 36 di tipo archeologico) si è scelto di scorrere i nodi con etichetta Opera, lanciando per ciascuno una query SPARQL sulle risorse Bene Culturale Artistico e Archeologico. I tempi dell'allineamento tra nodi del Property Graph e URI delle risorse RDF con un confronto di uguaglianza o di inclusione tra stringhe sono illustrati in tabella 3.

FASI SEPARATE				TOTALE			
classe ArCo	query SPARQL	elaboraz. dataset	# URI estratti	categoria	# nodi Graph	entity alignment	tempo totale di elaborazione
Agent / Person	3,4 s	35,9 s	7.260	:Artista	1.053	1.403,1 s	24 m 2 s
CulturalInstituteOrSite	2,1 s	7,4 s	1.096	:Istituzione	205	84,0 s	1 m 34 s
ArchitecturalHeritage	7,2 s	64,1 s	12.905	:Struttura	1.087	3.667,6 s	1 h 3 m 45 s
Site	7,7 s	78,4 s	5.538				
AddressArea	54,3 s	1,6 s	553	:Localita	359	14,8 s	1 m 11 s
City	7,1 s	0,1 s	360			11,5 s	19 s
FASE UNICA							
classe ArCo	categoria	# nodi Graph	allineamento e aggiornam. Neo4J	tempo totale di elaborazione			
HistoricOfArtisticProperty	:Opera	425	626,0 s	10 m 26 s			
ArcheologicalProperty	:Opera	58	510,6 s	8 m 31 s			

Tabella 3 – Tempi di elaborazione della fase di allineamento tra nodi K-MAG e URI

La tabella 4 e la figura 30 mostrano il risultato dell'allineamento con uno o più URI in valore assoluto e in percentuale. L'attribuzione allo stesso nodo di più URI non è sempre un errato allineamento tra entità che non corrispondono: infatti rientrano in questi casi sia l'associazione con URI relativi a Classi ontologiche diverse, sia le classificazioni multiple con schede duplicate per lo stesso bene o sito nel Catalogo dei Beni Culturali.

categoria	nodì	allineati	classe ArCo	1 uri	2 uri	>2 uri
:Artista	1.053	524 (50%)	Agent/Person	486 (93%)	35	3
:Opera	425	59 (14%)	HistoricOfArtisticProperty	29 (49%)	13	17
			ArcheologicalProperty	1 (25%)	1	2
:Struttura	1.086	428 (39%)	ArchitecturalHeritage/Site	115 (27%)	67	246
:Istituzione	205	76 (37%)	CulturalInstituteOrSite	47 (62%)	19	10
:Localita	359	101 (28%)	AddressArea	85 (84%)	8	8
			City	182 (77%)	46	9

Tabella 4 – Risultati dell'allineamento tra nodi K-MAG e URI

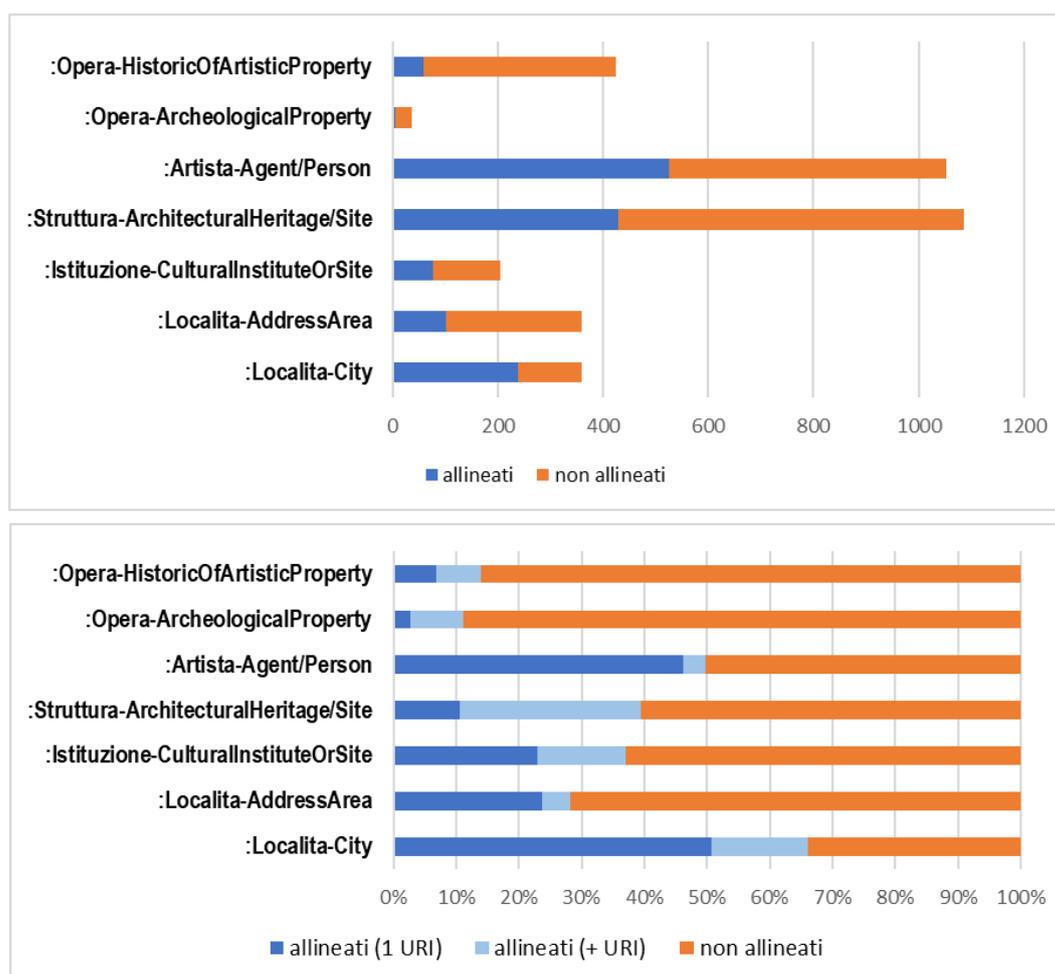


Figura 30 – Entità K-MAG allineate con URI in valore assoluto e percentuale

- La percentuale migliore riguarda l’allineamento tra **Località** e risorse City (66%), la classe che rappresenta unità amministrative di primo livello. Inferiore nel risultato è l’allineamento delle risorse AddressArea (52%), che rappresentano indirizzi con località e frazioni.
- Gli **Artisti** allineati sono circa la metà, con un’ottima percentuale di associazioni univoche (93% degli allineati). L’esito positivo è favorito dalla presenza di numerose varianti dei nomi degli artisti (nome e patronimico, nome e cognome, soprannome, nome e provenienza). Per diversi autori sono stati riscontrati in RDF molteplicità di URI per lo stesso autore.
- Le **Strutture** (edifici e monumenti) e le **Istituzioni culturali** (musei e altri istituti culturali) hanno una percentuale di allineamento più bassa, appena sopra il terzo di quelle presenti in K-MAG. La percentuale di associazione con URI univoci risulta migliore per le Istituzioni (62%) rispetto alle Strutture (40%), e la cosa è abbastanza prevedibile dato che di solito i Musei hanno nomi con meno varianti. Anche per i musei non mancano casi di risorse duplicate nel Catalogo dei Beni Culturali.
- Le **Opere** sono le entità per cui l’allineamento alle corrette URI è più problematico, data l’estrema variabilità e molteplicità delle denominazioni possibili per lo stesso Bene Artistico o Archeologico. A parte alcuni casi notevoli (come “La Muta” o la “Madonna di Senigallia”), sono spesso presenti denominazioni generiche (es. “Madonna col Bambino”, “Annunciazione”, “Crocifisso” o “kouros”) che ricorrono in opere tra loro distinte.

5.2.2 Raffinamento del confronto con stop-words e similarità

Per migliorare i risultati degli allineamenti tra entità in base alle loro denominazioni si sono applicate le funzioni di similarità descritte precedentemente non considerando nel confronto i segni di interpunzione e le *stop-words*, parole ricorrenti e generiche, come gli articoli e le preposizioni, che ostacolano il processo di allineamento.

La figura 31 riporta la lunghezza in parole delle diverse denominazioni di Autori, Istituti Culturali, Siti e Beni Architettonici in RDF Store, con la frequenza dei termini più ricorrenti tra i quali scegliere le *stop-words*. Alcune parole da scartare, comuni in diverse descrizioni, sono le seguenti:

['e', 'da', 'dal', 'di', 'del', 'della', 'dell', 'dei', 'delle', 'al', 'in']

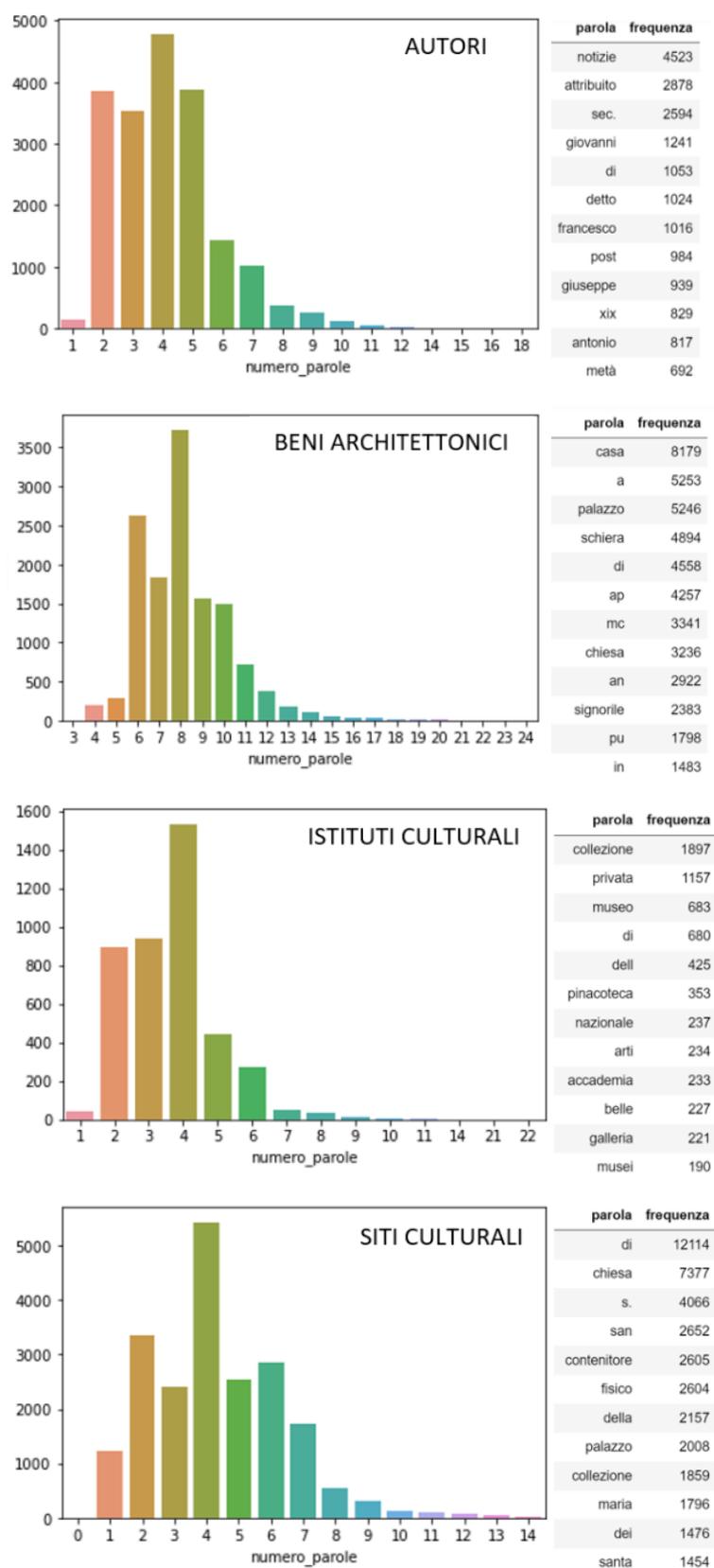


Figura 31 – Lunghezza in parole e frequenza dei termini nel sito dei Beni Culturali

Per la scelta della soglia minima per considerare positivo l'allineamento, la tabella 5 riporta come cresce la percentuale delle Istituzioni allineate in base alla soglia minima.

soglia	allineati	non allineati
Soglia 1,0	76 (37%)	129 (63%)
Soglia 0,8	92 (45%)	113 (55%)
Soglia 0,6	110 (54%)	95 (46%)
Soglia 0,4	153 (75%)	52 (25%)
Soglia 0,0	205 (100%)	-

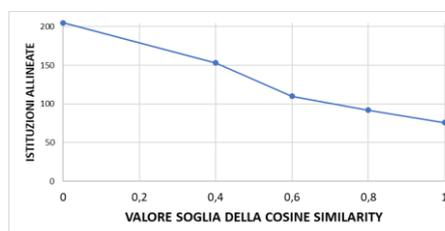


Tabella 5 – Risultati in base alla similarità dell'allineamento tra i nodi Istituzione e URI

Dopo aver definito *stop-words* specifiche per ogni categoria e aver scelto la soglia di similarità pari al valore 0,750, è stata modificata la funzione `AbbinaUriNodo` (algoritmo 3) con una chiamata della funzione `calcolo_similarita` (algoritmo 4). Questa funzione utilizza metodi della libreria `nltk` di Python per suddividere la frase in termini (*tokenize*) e calcolare la *cosine_distance*, dalla quale è immediato ricavare la *cosine_similarity*.

Algoritmo 3 – allineamento tra nodi del grafo e URI (versione 2)

```

1: function AbbinaUriNodo2(graph_conn, nomefile_uri, stopwords):
2:   graph ← Graph(graph_conn)
3:   list_nodi ← graph.run(MATCH pattern RETURN variables)
4:   df_uri ← read_csv(nomefile_uri)
5:   for nodo in list_nodi:
6:     for valore in nodo:
7:       for riga in df_uri:
8:         if valore.lower() = riga.lower():
9:           list_uri_ugual ← valore_uri
10:          if calcolo_similarita(valore, riga, stopwords) > soglia:
11:            list_uri_simil ← valore_uri

```

Algoritmo 4 – funzione di calcolo similarità tra frasi

```

1: function calcolo_similarita(testo1, testo2, stopwords):
2:   frase1 = nltk.tokenize(testo1).lower()
3:   frase2 = nltk.tokenize(testo2).lower()
4:   elenco_parole = set{frase1 union frase2}
5:   vettore1 = [0, ..., 0]
6:   vettore2 = [0, ..., 0]
7:   for parola in frase1:
8:     if parola in stopwords:
9:       continue
10:    vettore1[elenco_parole.index(parola)] += 1
11:   for parola in frase2:
12:     if parola in stopwords:
13:       continue
14:    vettore2[elenco_parole.index(parola)] += 1
15:   cosine_similarity = 1 - nltk.cosine_distance(vettore1, vettore2)
16:   return cosine_similarity

```

La tabella 6 illustra i tempi di elaborazione e i risultati del nuovo allineamento tra i nodi Artista, Istituto e Struttura del K-MAG e gli URI da RDF Store. Si nota un incremento dei tempi di allineamento dati per i confronti tra Artista e URI Autore e tra Struttura e URI Bene Architettonico, mentre risultano quasi invariati i tempi di confronto tra Istituzioni e URI Istituti Culturali e tra Strutture e URI Siti. La figura 32 confronta i risultati con l'allineamento su uguaglianza e inclusione di stringhe, e si nota quanto migliorano le percentuali di allineamento di Artisti (da 50% a 69%), Strutture (da 39% a 60%, con 647 allineamenti) e Istituzioni (da 37% a 52%).

FASI SEPARATE			FASE II			TOTALE	
classe ArCo	query SPARQL	elaboraz. dataset	# URI estratti	categoria	# nodi Graph	entity alignment	tempo totale di elaborazione
Agent / Person	5,6 s	26,0 s	7.332	:Artista	1.053	15.887,4 s	4 h 25 m 19 s
CulturalInstituteOrSite	7,7 s	7,0 s	1.088	:Istituzione	205	58,6 s	1 m 13 s
ArchitecturalHeritage	9,4 s	68,2 s	12.906	:Struttura	1.086	11.853,4 s	3 h 18 m 51 s
Site	20,2 s	52,9 s	6.209	:Struttura	1.086	4.582,9 s	1 h 17 m 36 s

RISULTATI DELL'ALLINEAMENTO							
categoria	nodì	allineati	classe ArCo	1 uri	2 uri	3 uri	>3 uri
:Artista	1.053	727 (69%)	Agent/Person	343 (47%)	204 (28%)	87	93
:Struttura	1.086	351 (32%)	ArchitecturalHeritage	150 (43%)	53 (15%)	29	119
		478 (44%)	Site	247 (52%)	116 (24%)	58	57
:Istituzione	205	107 (52%)	CulturalInstituteOrSite	67 (63%)	24 (22%)	9	7

Tabella 6 – Tempi di elaborazione e risultati dell'allineamento con funzione di similarità

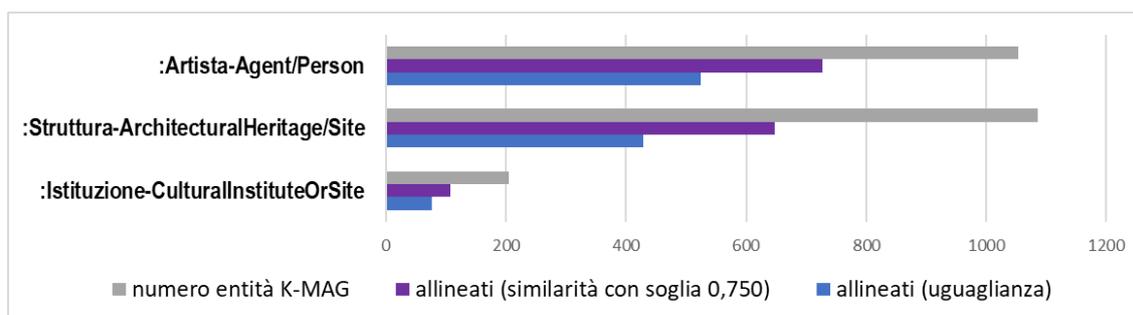


Figura 32 – Confronto tra allineamento con uguaglianza e con similarità

La percentuale di associazione con URI univoco rimane la stessa per le Istituzioni (da 62% a 63% delle allineate), aumenta per le Strutture (effetto che dipende fortemente dalla separazione tra Beni Architettonici e Siti) e cala vistosamente per gli Artisti da 93% a 47% (calo assoluto da 486 a 343). Questo effetto richiede una verifica approfondita, per capire se si tratta di attribuzioni errate che associano URI di persone diverse con nomi simili oppure se dal confronto emergono altri URI associati agli stessi artisti.

Si potrebbe allargare il campo del processo di Entity Alignment incrociando i dati biografici, le opere realizzate, i periodi di esistenza e di attività. In generale per un efficace allineamento è essenziale prevedere metriche opportune, tarare la soglia minima di similarità e verificare con cura i risultati.

Al termine degli allineamenti, lo script Python aggiorna K-MAG, aggiungendo ai nodi proprietà multi-valore con la lista delle URI recuperate da RDF Store dei Beni Culturali. La figura 33 mostra il nodo che rappresenta la Mole Vanvitelliana su Neo4J Browser, con archi e con proprietà aggiornate al termine dell'allineamento.

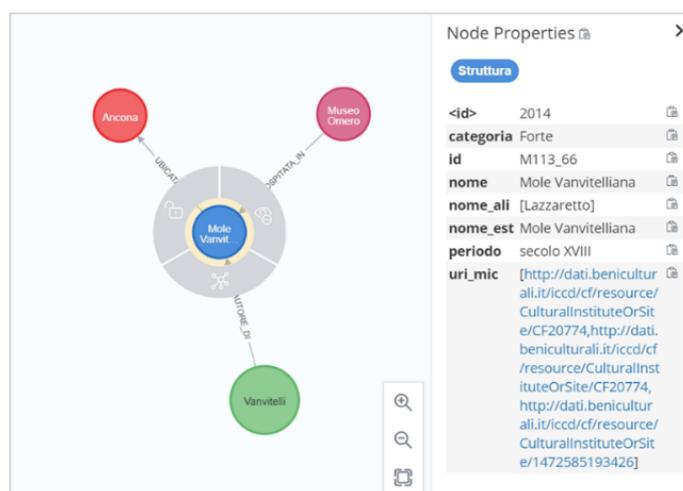


Figura 33 – Nodo del K-MAG relativo alla Mole Vanvitelliana con i valori multiproprietà URI

5.2.3 Macrofase II: integrazione dati su entità già presenti in K-MAG

L'estrazione di informazioni interessanti relative alle opere, alle strutture e alle istituzioni culturali presenti in K-MAG, risulta più semplice per quei nodi per i quali è stato individuato un riferimento unico. In questo caso è sufficiente interrogare l'endpoint SPARQL con questo parametro per ottenere risultati da cui estrarre i dati interessanti per l'integrazione delle informazioni contenute in K-MAG. Dato che le località del dominio di interesse sono già state precaricate in K-MAG, le elaborazioni in questa macro-fase riguardano quattro categorie di nodi (Opere, Artisti, Istituzioni e Strutture). Questa fase permette anche di effettuare verifiche a posteriori sulle associazioni tra nodi e URI effettuate nella prima macro-fase.

La funzione `AbbinaDati` (algoritmo 5) estrae l'elenco dei nodi della categoria di interesse contenenti URI e richiama la funzione `confronta` (algoritmo 6) per allineare i valori delle proprietà dei nodi presenti in K-MAG con le informazioni sul Catalogo dei Beni Culturali collegate alla risorsa identificata dall'URI.

Algoritmo 5 – allineamento tra nodi e risorse in base alle corrispondenti URI

```

1: function AbbinaDati(endpoint, source_conn)
2:   sparql ← SPARQLWrapper(endpoint)
3:   graph ← Graph(source_conn)
4:   list_nodi ← graph.run(MATCH n:Label WHERE uri IS NOT NULL RETURN id)
5:   for id_nodo in list_nodi:
6:     dati_nodo ← graph.run(MATCH pattern RETURN variables)
7:     for uri in dati_nodo.list_uri
8:       sparql.setQuery(SELECT variables WHERE {pattern_match FILTER uri})
9:       result ← sparql.query()
10:      for result in result.bindings:
11:        dict_dati_risorsa ← result[result.variable].value
12:        if exists(prop_nodo):
13:          confronta(id_nodo,prop_nodo,dict_dati_risorsa[prop])
14:          df_confronti ← esito_confronto_prop
15:      aggiorna_grafo()
16:      df_confronti.to_csv(nomefile)

```

Algoritmo 6 – funzione di confronto tra proprietà e collezioni di valori

```

1: function confronta(id, prop, valore, collezione):
2:   confr1 = 0
3:   confr2 = 0
4:   if valore.lower() in [x.lower() for x in collezione]:
5:     confr1 = 1
6:   for item in collezione:
7:     testo1 = valore.lower()
8:     testo2 = item.lower()
9:     confr2 = max(confr2, calcolo_similarita(testo1, testo2))
10:  return [id, prop, valore, collezione, confr1, confr2]

```

La tabella 7 mostra i tempi dell'allineamento effettuato con funzione di similarità tra i valori delle proprietà presenti nel nodo K-MAG e i valori *literal* scelti tra quelli presenti nelle triple collegate alle corrispondenti risorse.

Categoria K-MAG	nodi con URI	classi ArCo confrontate	tempo di allineamento
:Opera	63	HistoricOfArtisticProperty	163,3 s
		ArcheologicalProperty	64,9 s
:Artista	524	Agent / Person	198,1 s
:Istituzione	76	CulturalInstituteOrSite	43,7 s
:Struttura	428	ArchitecturalHeritage	509,1 s
		Site	698,0 s

Tabella 7 – Tempi di elaborazione dei confronti tra proprietà K-MAG e dati RDF

I risultati del confronto sono riportati in tabella 8 come valori medi della funzione di similarità, distinti base alla coppia proprietàGraphNode/risorsaRDF confrontata.

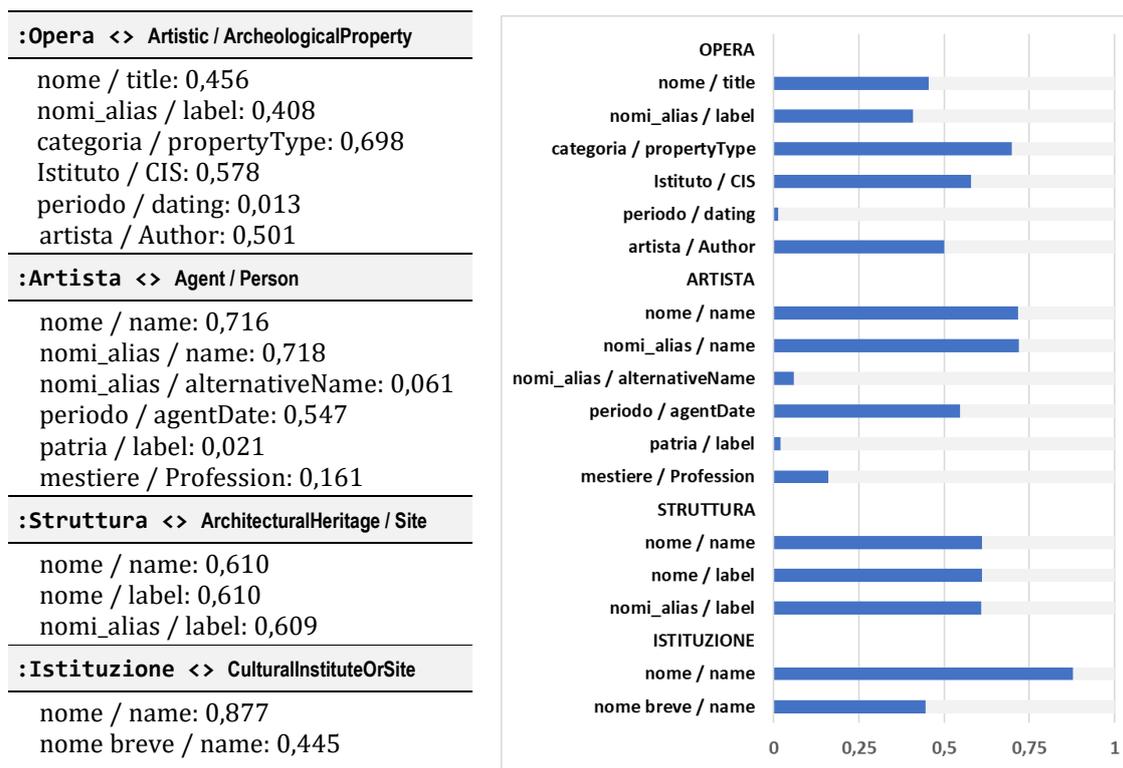


Tabella 8 – Valori medi della funzione di similarità nei confronti tra le singole proprietà

Per approfondire le criticità di questo processo, si mostra un esempio concreto di integrazione tra K-MAG e RDF Store, seguendo la catena degli allineamenti. Si parte dal nodo del grafo K-MAG che rappresenta il dipinto di Raffaello conosciuto come “La Muta” (figura 34) custodito nel Palazzo Ducale di Urbino. Grazie alla sua denominazione particolare, nella prima macro-fase l’opera era stata associata ad un URI univoco. Con una interrogazione sull’end point SPARQL è immediato acquisire informazioni relative al dipinto, al suo autore, al Museo e al Palazzo che lo contiene.



Figura 34 – Il dipinto “La Muta” di Raffaello e il suo contenitore, il Palazzo Ducale di Urbino

Il primo allineamento riguarda le proprietà associate all'opera: la tabella 9 riporta gli esiti del confronto di uguaglianza (Confr1) e del confronto tramite funzione di similarità (Confr2). Si nota che la denominazione 'La Muta' provoca un allineamento perfetto, mentre nomi più complessi (es. 'La Muta, ritratto di donna (dipinto) di Sanzio Raffaello') richiedono da un lato l'estrazione di termini chiave, dall'altro un vocabolario che considerasse equivalenti 'gentildonna' e 'donna'. Un confronto interessante è quello tra le date di realizzazione dell'opera: '1507' e 'ca.1505-ca.1509', un termine puntuale e un intervallo con estremi non certi. Una persona umana li può tranquillamente ritenere pressoché equivalenti, considerando l'incertezza delle datazioni nei secoli passati, ma un algoritmo ha bisogno di regole per trattare questi casi.

IdOpera	Informazione	K-MAG	RDF store	Confr1	Confr2
148	nome - title	La Muta	[La Muta]	1	1.000000
148	nome - label	La Muta	[La Muta, ritratto di donna (dipinto) di Sanzi...	0	0.426401
148	nomi alias - label	[Ritratto di gentildonna]	[La Muta, ritratto di donna (dipinto) di Sanzi...	0	0.480384
148	periodo - dating	1507	[ca 1505 - ca 1509]	0	0.000000
148	categoria - propertyType	Dipinto	[Dipinto, DIPINTO, dipinto]	1	1.000000
148	autore - Author	[https://w3id.org/arco/resource/Agent/c2fa7805...	[https://w3id.org/arco/resource/Agent/1588314f...	1	1.000000
148	istituto - CIS	[https://w3id.org/arco/resource/Culturallnstit...	[https://w3id.org/arco/resource/Culturallnstit...	1	1.000000
148	nome luogo istit. - Address	Urbino	[PU, Urbino, Italia, Marche, PU, Urbino]	0	0.707107

Tabella 9 – Confronto delle proprietà relative al dipinto "La Muta"

La tabella 10 riporta i confronti relativi all'autore dell'opera, l'urbinate Raffaello. C'è un allineamento perfetto sia per il nome del pittore, presente in diverse forme ('Raffaello', 'Raffaello Sanzio', 'Raffaello Santi'), sia per il periodo di esistenza, grazie all'algoritmo che scarta preventivamente i segni di interpunzione.

IdAutore	Informazione	K-MAG	RDF store	Confr1	Confr2
RAFFAELLO	nome	Raffaello Sanzio	[Sanzio, Raffaello - 1483/ 1520, Raffaello San...	1	1.000000
RAFFAELLO	nome alternativo	[Raffaello, Raffaello Santi, Santi Raffaello]	[Raffaello, Santi Raffaello]	1	1.000000
RAFFAELLO	periodo	1483 - 1520	[1483/ 1520]	1	1.000000
RAFFAELLO	mestiere	Pittore e Architetto	[https://w3id.org/arco/resource/Profession/pit...	0	0.000000
RAFFAELLO	patria	da Urbino	[Sanzio, Raffaello - 1483/ 1520, Raffaello San...	0	0.000000

Tabella 10 – Confronto delle proprietà relative all'artista Raffaello Sanzio

L'aggiornamento dei dati relativi al Museo che custodisce il dipinto (“Galleria Nazionale delle Marche”) e all’Edificio che lo ospita fisicamente (“Palazzo Ducale”) appare a prima vista più semplice, ma si complica per il fatto che seguendo la catena di triple RDF si ottengono 8 URI di tipo `CulturalInstituteOrSite` per l’entità “Galleria Nazionale delle Marche” e 5 URI di tipo `Site` per l’entità “Palazzo Ducale di Urbino”. Alcune di queste URI contengono dati simili ma non identici e occorre scegliere con cura quali informazioni sono utili per K-MAG.

URI Istituto Culturale “Galleria Nazionale delle Marche”:

```
'https://w3id.org/arco/resource/CulturalInstituteOrSite/89edbd614207af77fbd2322ca0b79ae5'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/85c064434f679a7a2e7510f0df518f76'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/df90e67dfa8369bf292933f326e0ae7a'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/8705667f00b84214c2e7e5b43ffe0e08'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/f4ef42db2bb93e4466b3a79cb7b36760'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/c5d394f9e7f4c1282934c956c7f45d9c'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/9aec82e39da6f1dd8cd57cda9d295202'
'https://w3id.org/arco/resource/CulturalInstituteOrSite/DBunicoCG20392'
```

URI Sito “Palazzo Ducale di Urbino”:

```
'https://w3id.org/arco/resource/Site/cdb9160b494c2069d14b15d1b08db835'
'https://w3id.org/arco/resource/Site/40d2e3519f5f834e0c36251e7521902e'
'https://w3id.org/arco/resource/Site/5c466728dadd6189288a0acf7193e649'
'https://w3id.org/arco/resource/Site/961b5fa91a5cdb998337170221e25a31'
'http://dati.beniculturali.it/iccd/cf/resource/CulturalInstituteOrSite/CF20392'
```

Questo esempio mostra quanto non sia banale il processo di Entity Alignment, anche quando sono noti gli identificativi URI. È necessario conoscere il dominio di interesse, nonché i criteri e le modalità di classificazione, spesso sovrapposti e stratificati nel tempo. Occorre saper distinguere i contenuti simili, le risorse che descrivono la stessa cosa sotto differenti punti di vista e le denominazioni uguali associate ad entità tra loro differenti, il tutto per produrre algoritmi che riescano ad eseguire queste azioni meglio di quanto sappia fare una persona umana.

5.2.4 Macrofase III: integrazione dati con nuove entità in K-MAG

La strategia più promettente di integrazione tra dati registrati in K-MAG e contenuto informativo dalle Schede dei Beni Culturali inizia estraendo gli elenchi delle opere custodite nei Musei con i relativi autori (figura 35).



Figura 35 – Estrazione dell’elenco delle opere custodite nei Musei

Le figure 36 e 37 mostrano i codici SPARQL utilizzati nello script Python per estrarre gli elenchi delle Istituzioni Culturali della regione storico-geografica marchigiana, con il relativo numero di Beni Culturali di tipo Archeologico e Artistico.

```

SELECT DISTINCT ?uri_museo, ?nome_museo, ?luogo,
COUNT(DISTINCT ?bc) AS ?numero WHERE {
  ?bc rdf:type arco:ArchaeologicalProperty ;
  arco:regionIdentifier ?regione ;
  a-loc:hasCulturalPropertyAddress/clvapit:hasCity/rdfs:label ?luogo ;
  a-loc:hasCulturalInstituteOrSite ?uri_museo .
  ?uri_museo rdfs:label ?nome_museo .
  FILTER (?regione = "11" || REGEX(?luogo, "Gubbio", "i") )
} ORDER BY DESC(?numero)

```

	uri_museo	nome_museo	luogo	numero
0	https://w3id.org/arco/resource/CulturalInstitu...	Museo Archeologico Nazionale delle Marche	Ancona	10472
1	https://w3id.org/arco/resource/CulturalInstitu...	Museo archeologico Nazionale delle Marche	Ancona	10472
2	https://w3id.org/arco/resource/CulturalInstitu...	Museo civico archeologico "Cesare Cellini"	Ripatransone (?)	990
3	https://w3id.org/arco/resource/CulturalInstitu...	Museo Civico Archeologico "Cesare Cellini"	Ripatransone	990
4	https://w3id.org/arco/resource/CulturalInstitu...	Museo Civico Archeologico "Cesare Cellini"	Ripatransone (?)	990
...
70	https://w3id.org/arco/resource/CulturalInstitu...	Museo civico archeologico " Cesare Cellini"	Ripatransone	1
71	https://w3id.org/arco/resource/CulturalInstitu...	Museo Archeologico Nazionale	Ascoli Piceno	1
72	https://w3id.org/arco/resource/CulturalInstitu...	Museo Archeologico Nazionale delle Marche	Ancona	1
73	https://w3id.org/arco/resource/CulturalInstitu...	Museo Archeologico Statale	Acquaviva Picena	1

Figura 36 – Query SPARQL e risultato con l'elenco dei Musei per numero di beni archeologici

```

SELECT DISTINCT ?uri_museo, ?nome_museo, ?luogo,
COUNT(DISTINCT ?bc) AS ?numero WHERE {
  ?bc rdf:type arco:HistoricOrArtisticProperty ;
  arco:regionIdentifier ?regione ;
  a-loc:hasCulturalPropertyAddress/clvapit:hasCity/rdfs:label ?luogo ;
  a-loc:hasCulturalInstituteOrSite ?uri_museo .
  ?uri_museo rdfs:label ?nome_museo .
  FILTER (?regione = "11" || REGEX(?luogo, "Gubbio", "i"))
} ORDER BY DESC(?numero)

```

	uri_museo	nome_museo	luogo	numero
0	https://w3id.org/arco/resource/CulturalInstitu...	Pinacoteca Civica	Ascoli Piceno	1853
1	https://w3id.org/arco/resource/CulturalInstitu...	Museo Civico	Urbania	1775
2	https://w3id.org/arco/resource/CulturalInstitu...	Pinacoteca e Musei Civici	Macerata	1022
3	https://w3id.org/arco/resource/CulturalInstitu...	Galleria Nazionale delle Marche di Urbino	Urbino	936
4	https://w3id.org/arco/resource/CulturalInstitu...	Museo Colocci	Jesi	900
...
96	https://w3id.org/arco/resource/CulturalInstitu...	Musei Civici	Pesaro	1
97	https://w3id.org/arco/resource/CulturalInstitu...	Pinacoteca Civica	Ascoli Piceno	1

Figura 37 – Query SPARQL e risultato con l'elenco dei Musei per numero di beni artistici

Per produrre un elenco dei Beni Culturali da cui estrarre le informazioni utili, fare il confronto con quanto è presente in K-MAG e procedere con la fase di integrazione, si è considerato un sottoinsieme dei Musei. Per i Istituti contenenti beni archeologici sono state selezionate le URI riconducibili al Museo Archeologico Nazionale delle Marche, per quelli contenenti beni artistici sono state selezionate le URI dei seguenti Musei ubicati nelle città che sono state capoluogo di Provincia o Delegazione:

- Galleria Nazionale delle Marche (Urbino)
- Musei Civici (Pesaro)
- Museo delle Ceramiche (Pesaro)
- Pinacoteca Francesco Podesti (Ancona)
- Pinacoteca e Musei Civici (Macerata)
- Pinacoteca e Biblioteca Mozzi-Borgetti (Macerata)
- Pinacoteca Civica (Camerino)
- Pinacoteca Comunale (Fermo)
- Galleria di Arte Contemporanea Osvaldo Licini (Ascoli Piceno)
- Pinacoteca Civica (Ascoli Piceno)

In figura 38 è mostrato il codice SPARQL per richiedere all'endpoint SPARQL del Catalogo dei Beni Culturali l'elenco dei beni che risultano custoditi all'interno del Museo Archeologico Nazionale delle Marche. Lo script ha estratto in 6,8 secondi 21.808 record che corrispondono a 10.904 URI distinti. Escludendo le distinzioni case-sensitive ai beni archeologici corrispondono 4.164 *label* distinte, risultato plausibile per manufatti spesso identificati con denominazioni generiche come 'vaso' o 'conchiglia forata'.

```

SELECT DISTINCT ?bc, ?label, ?cis
  WHERE { ?bc rdf:type arco:ArchaeologicalProperty ;
          rdfs:label ?label ;
          a-loc:hasCulturalInstituteOrSite ?cis
          FILTER ( ?cis = < uriMuseo1 > || < uriMuseo2 > || ... ) }

```

bc	label	cis
https://w3id.org/arco/resource/ArchaeologicalP...	kylix - produzione attica (prima metà sec. V a.C)	https://w3id.org/arco/resource/CulturalInstitu...
https://w3id.org/arco/resource/ArchaeologicalP...	kylix - produzione attica (prima metà sec. V a.C)	https://w3id.org/arco/resource/CulturalInstitu...
https://w3id.org/arco/resource/ArchaeologicalP...	anello da sospensione (sec. VI)	https://w3id.org/arco/resource/CulturalInstitu...
https://w3id.org/arco/resource/ArchaeologicalP...	anello da sospensione (sec. VI)	https://w3id.org/arco/resource/CulturalInstitu...
https://w3id.org/arco/resource/ArchaeologicalP...	lekane - tardo stile libero (ultimo quarto sec...	https://w3id.org/arco/resource/CulturalInstitu...

Figura 38 – Codice SPARQL per ottenere le opere di un Museo Archeologico

In figura 39 è mostrato il codice SPARQL per richiedere l'elenco dei beni artistici custoditi nei Musei selezionati nelle città capoluogo. In 28,8 secondi sono stati estratti 25.188 record corrispondenti a 5.256 Beni Artistici distinti per URI, con i relativi autori. Escludendo le distinzioni case-sensitive e le versioni diverse per lingua, a questi Beni sono associate 4.494 label distinte. È evidente che denominazioni identiche si riferiscono ad URI e a beni distinti.

```

SELECT DISTINCT ?bc, ?label, ?author, ?lauthor, ?cis, ?lcis
WHERE { ?bc rdf:type arco:HistoricOrArtisticProperty ;
        rdfs:label ?label ;
        a-cd:hasAuthor ?author ;
        a-cd:hasAuthor/rdfs:label ?lauthor ;
        a-loc:hasCulturalInstituteOrSite ?cis .
        ?cis rdfs:label ?lcis
FILTER ( ?cis = < uriMuseo1 > || < uriMuseo2 > || ... ) }

```

bc	label	author	lauthor	cis	lcis
rArtis...	gioco dello scappagatto (disegno) by Podesti F...	https://w3id.org/arco/resource/Agent/c019044e6...	Podesti Francesco - 1800/1895	https://w3id.org/arco/resource/CulturalInstitu...	Pinacoteca Civica F. Podesti
rArtis...	gioco dello scappagatto (disegno) di Podesti F...	https://w3id.org/arco/resource/Agent/c019044e6...	Podesti Francesco - 1800/1895	https://w3id.org/arco/resource/CulturalInstitu...	Pinacoteca Civica F. Podesti
rArtis...	Cristo trasportato al sepolcro e cinque angeli...	https://w3id.org/arco/resource/Agent/f0f94283f...	Zuccari Taddeo (e Aiuti)	https://w3id.org/arco/resource/CulturalInstitu...	Galleria Nazionale delle Marche
rArtis...	Madonna con Bambino e santi (dipinto,	https://w3id.org/arco/resource/Agent/cf28f8061...	De Magistris Simone (e Aiuti)	https://w3id.org/arco/resource/CulturalInstitu...	Galleria Nazionale delle Marche

Figura 39 – Codici SPARQL per ottenere l'elenco dei beni artistici con gli autori

Gli artisti autori dei beni artistici elencati sono individuati da 1.809 URI da 3.156 label distinti (3.145 se la verifica è case-insensitive). Le coppie distinte (URI, label) sono 3.839, quindi diversi artisti corrispondono a più di una URI. Diverse duplicazioni dipendono da attribuzioni non condivise o riferite alla bottega dell'autore.

Il passo successivo comporta il confronto con quanto è presente nel grafo K-MAG. L'allineamento tra i nodi Istituzione in K-MAG e gli URI delle risorse di tipo Istituto Culturale in base alla corrispondenza delle denominazioni produce i risultati mostrati in tabella 11. Si nota che per ogni Museo risultano URI molteplici e che quelli allineati sono 23 su 34. Il mancato allineamento deriva dalla presenza di termini diversi dal significato simile (Nazionale/Statale, Comunale/Civica) da gestire con un vocabolario di sinonimi adeguato.

ISTITUTO CULTURALE	Risorse URI in RDFStore	URI allineati	URI da allineare
Museo Archeologico Nazionale delle Marche (Ancona)	5	4	1
Galleria Nazionale delle Marche (Urbino)	5	5	-
Musei Civici (Pesaro)	7	7	-
Museo delle Ceramiche (Pesaro)	1	1	-
Pinacoteca Francesco Podesti (Ancona)	2	2	-
Pinacoteca comunale e Musei Civici (Macerata)	2	-	2
Pinacoteca e Biblioteca Comunale Mozzi-Borgetti (Macerata)	1	1	-
Pinacoteca Civica e Museo Archeologico (Camerino)	2	2	-
Pinacoteca Comunale (Fermo)	2	-	2
Galleria di Arte Contemporanea O.Licini (Ascoli Piceno)	1	1	-
Pinacoteca Civica (Ascoli Piceno)	6	-	6

Tabella 11 – URI associati agli Istituti culturali selezionati

Il secondo allineamento riguarda gli elenchi degli artisti che hanno opere nei Musei. La tabella 12 riporta il numero di Author collegati agli Istituti Culturali scelti e il numero di Artisti collegati nel grafo K-MAG alle stesse Istituzioni con archi [:HA_OPERA_IN]. Il numero di entità allineate è discordante quando gli Artisti sono stati associati a più URI.

ISTITUTO CULTURALE	URI Author collegati RDFStore	URI Author allineati	Artisti collegati in K-MAG [:HA_OPERA_IN]	Artisti collegati in K-MAG allineati
Galleria Nazionale delle Marche	447	32	68	32 (47%)
Pinacoteca e Musei Civici di Macerata	416	11	18	11 (61%)
Musei Civici di Pesaro	293	1	11	1 (9%)
Pinacoteca Civica di Ascoli	257	59	154	58 (38%)
Galleria di Arte Contemporanea Licini	208	2	2	2 (100%)
Pinacoteca Francesco Podesti	208	34	47	34 (72%)
Pinacoteca Comunale di Fermo	45	6	12	6 (50%)
Pinacoteca Comunale Mozzi-Borgetti	41	-	-	-
Museo delle Ceramiche di Pesaro	33	1	2	1 (50%)
Pinacoteca Civica di Camerino	17	-	15	-

Tabella 12 – Collegamenti tra Artisti e Istituti in K-MAG allineati con URI da RDF Store MiC

La tabella 13 riporta il numero di beni del Catalogo dei Beni Culturali collegati alle Istituzioni scelte e il numero di Opere collegate nel grafo K-MAG alle stesse Istituzioni con archi [:CUSTODITA_IN], con scarse percentuali di abbinamento⁸.

⁸ In particolare l'allineamento tra denominazioni di beni archeologici è poco accurato: "collare" e "kouros" sono associati a URI che identificano una serie di reperti diversi.

ISTITUTO CULTURALE	URI Property RDFStore	URI allineati a Opere	Opere collegate in K-MAG [:CUSTODITA_IN]	Opere colleg. in K-MAG allineate
Museo Archeologico Nazionale delle Marche	10.472	15	26	2 (8%)
Galleria Nazionale delle Marche	1.299	34	81	21 (26%)
Pinacoteca Civica di Ascoli	1.297	7	12	1 (8%)
Galleria di Arte Contemp. Licini	748	-	-	-
Pinacoteca e Musei Civici di Macerata	719	-	1	-
Musei Civici di Pesaro	464	-	5	-
Pinacoteca Francesco Podesti	414	-	4	-
Pinacoteca Comunale di Fermo	108	1	7	1 (14%)
Pinacoteca Civica di Camerino	75	-	5	-
Museo delle Ceramiche di Pesaro	74	-	-	-
Pinacoteca Com. Mozzi-Borgetti	58	-	-	-

Tabella 13 – Collegamenti tra Opere e Istituti in K-MAG allineati con URI da RDF Store MiC

Anche l'allineamento dei nodi a URI di beni artistici non dà risultati apprezzabili. Considerando i beni artistici e i relativi autori, le Opere collegate in K-MAG con Artisti tramite archi [:ESEGUITA_DA] allineate a coppie di URI che individuano risorse Author e HistoricOrArtisticProperty legate da predicati isAuthorOf sono molto poche:

- Galleria Nazionale Marche* - Crivelli Vittore: Pietà, 2 URI
 - Giovanni Antonio da Pesaro: Crocifissione, 2 URI
 - Piero della Francesca: Madonna di Senigallia, 1 URI
 - Pietro da Rimini: Crocifisso, 3 URI
 - Puccio di Simone: Madonna con Bambino e Santi, 4 URI
 - Raffaellino del Colle: Madonna del Soccorso e Santi, 1 URI
 - Raffaello: La Muta, 1 URI
 - Santi Giovanni: Pala Buffi, 1 URI
 - Signorelli Luca: Crocifissione, 2 URI
 - Zuccari Federico: Cristo trasportato al sepolcro, 1 URI
- Pinacoteca di Ascoli Piceno* - Cola dell'Amatrice: Madonna, 7 URI

Alcune di queste opere sono allineate ad un'unica risorsa, altre opere sono collegate ad URI distinte relative alla stessa opere o ad opere distinte⁹.

⁹ Evidente è il caso di Cola dell'Amatrice, il cui dipinto "Madonna" custodito alla Pinacoteca di Ascoli è stato messo in corrispondenza con alcune sue opere presenti nello stesso Museo e che trattano lo stesso soggetto. Non essendoci per quel periodo denominazioni ufficiali e univoche, le opere vengono distinte con appellativi o dettagli della scena: nel caso di Cola i sette dipinti distinti sono definiti "Madonna con il Bambino", "Madonna del Buon Amore", "Madonna della Cesta", "Madonna della Seggiola", "Madonna del Riparo", "Madonna del Rosario", "Madonna di Loreto".

Al termine di questa fase si è ripetuto l'allineamento tra nodi Istituzione in K-MAG e Istituti Culturali in RDF Store, considerando tutti gli 80 Istituti Culturali, presenti nella regione geografica di interesse, che custodiscono beni artistici riconducibili ad autori. L'associazione di URI eseguita nella prima macro-fase con una soglia di similarità 0,750 aveva allineato 44 entità su 80. Se si abbassa la soglia della funzione di similarità da 0,750 a 0,500, si ottengono i risultati mostrati in tabella 14.

index1	nome1	luogo1	index2	nome2	luogo2	simil
18	museo delle memorie patrie pollentine	pollenza	143	pinacoteca e museo delle patrie memorie	pollenza	0.750000
13	pinacoteca civica	fabriano	110	pinacoteca civica b. molajoli	fabriano	0.707107
14	museo civico pinacoteca scipione gentili	san ginesio	177	museo pinacoteca s. gentili	san ginesio	0.670820
33	museo comunale	potenza picena	120	pinacoteca comunale	potenza picena	0.500000
17	pinacoteca civica	jesi	98	pinacoteca comunale	jesi	0.500000
10	pinacoteca civica	jesi	98	pinacoteca comunale	jesi	0.500000
9	pinacoteca civica	jesi	98	pinacoteca comunale	jesi	0.500000
19	pinacoteca colonica	porto recanati	119	pinacoteca comunale	porto recanati	0.500000
23	pinacoteca civica	ascoli piceno	191	pinacoteca comunale	ascoli piceno	0.500000
6	pinacoteca comunale	fermo	184	biblioteca comunale	fermo	0.500000
6	pinacoteca comunale	fermo	182	pinacoteca civica	fermo	0.500000
27	biblioteca comunale	ostra	90	pinacoteca comunale	ostra	0.500000
29	pinacoteca civica	ascoli piceno	191	pinacoteca comunale	ascoli piceno	0.500000
32	pinacoteca civica	ascoli piceno	191	pinacoteca comunale	ascoli piceno	0.500000
1	pinacoteca civica	ascoli piceno	191	pinacoteca comunale	ascoli piceno	0.500000
0	museo civico	urbania	57	museo arcidiocesano	urbania	0.500000
12	pinacoteca civica	ascoli piceno	191	pinacoteca comunale	ascoli piceno	0.500000

Tabella 14 – Confronto per similarità con soglia minima 0,500 tra Istituzioni K-MAG e URI

Esaminando con attenzione il risultato delle 17 associazioni proposte, si nota che gli allineamenti con soglia superiore al minimo sono corretti, ma sui restanti 14 allineamenti con soglia 0,500 ben 5 sono imprecisi. Si tratta di frasi con sole due parole di cui una corrispondente e una diversa, da confermare valutandone la semantica o il contesto. In generale qualunque attività di Entity Alignment che parte da analisi del testo deve saper leggere tra le righe, esplorando le relazioni ed esaminando i dati delle entità collegate, per riuscire a confermare con certezza probabilistica la correttezza di ciascun allineamento.

Dopo avere estratto gli elenchi di tutti i Beni culturali con dei relativi autori custoditi all'interno degli Istituti Culturali, è possibile avviare la terza macro-fase del processo di integrazione del Knowledge Graph K-MAG evidenziata in figura 40.

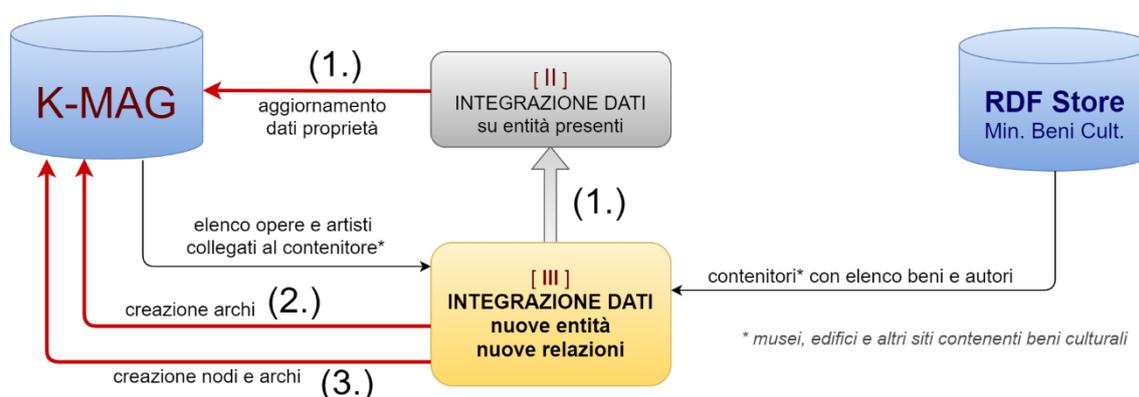


Figura 40 – Terza macro-fase nel workflow del processo di integrazione K-MAG

I passi principali da effettuare in successione in questa macro-fase sono i seguenti:

1. Partendo dai nodi Artista associati con URI e collegati alle Istituzioni con gli archi [:HA_OPERA_IN], si aggiornano i valori delle proprietà ripetendo le operazioni della seconda macro-fase. La stessa operazione viene effettuata per i nodi Opera associati con URI e collegati ad archi [:ESEGUITA_DA] e [:CUSTODITA_IN]. In questo caso le duplicazioni e le incertezze di attribuzione aumentano, occorre fare attenzione per evitare aggiornamenti incompleti o errati. Invece di aggiornare i valori delle proprietà presenti, può essere preferibile aggiungere nuove proprietà specifiche da confrontare successivamente, con il vantaggio di poter ricaricare dati LOD a cadenza predefinita.
2. Si verificano quali sono i nodi Autore e Opera con URI associati per cui non risultano archi [:ESEGUITA_DA], [:CUSTODITA_IN] e [:HA_OPERA_IN] e si creano nuovi archi attribuendo proprietà in base ai dati presenti in RDF Store, con la necessaria accuratezza per evitare di inserire collegamenti non reali.
3. Si considerano tutti i Beni artistici e archeologici i cui URI non sono stati abbinati con entità presenti in K-MAG e si effettuano le seguenti operazioni:
 - a) si verifica preventivamente se per ciascun Museo o Ente contenitore esistano opere e relativi autori con denominazioni simili, per evitare di creare nuovi nodi riconducibili ad entità già presenti;
 - b) in caso di riscontro positivo, si effettuano le operazioni al punto 1. con la dovuta attenzione nel trattare le diverse denominazioni;
 - c) per le opere e gli autori non presenti in K-MAG, si estraggono da RDF Store tutte le informazioni interessanti da caricare, si selezionano i valori delle proprietà e si aggiorna il grafo K-MAG, creando i nuovi nodi e i nuovi archi con le proprietà relative.

5.3 Tecniche avanzate per l'integrazione

Le operazioni volte ad incrementare il patrimonio informativo del Knowledge Graph comportano una serie di controlli molto accurati. Il problema principale è assicurare la qualità dell'integrazione dei contenuti, e questo comporta una "comprensione" del contenuto per assicurare l'uniformità ed eliminare quanto possibile le ambiguità. È possibile effettuare elaborazioni supervisionate da soggetti che hanno la conoscenza del dominio e sfruttando tecniche avanzate di *deep learning*, con algoritmi in grado di operare riconoscimento semantico.

Strumenti avanzati di analisi testuale e delle immagini possono essere integrati nella fase di Entity Alignment delle opere artistiche, archeologiche e architettoniche, per risolvere il problema fondamentale del riconoscimento dell'identità e della risoluzione delle ambiguità. Esistono tecniche molto interessanti utilizzate in *computer vision* che comportano l'analisi delle immagini e ricostruzioni in modelli 3D.

In effetti la natura del dominio degli oggetti culturali è principalmente visiva, in fondo i beni culturali sono oggetti nati per essere visti. Quasi nessuno, ascoltando un testo scritto o una descrizione orale, saprebbe riconoscere di quale opera si tratta oppure distinguere due opere distinte. Però molti sono in grado di riconoscere un quadro o un edificio con una rapida occhiata e di capire facilmente dall'esame di riproduzioni stampate o digitali se si tratta della stessa opera oppure di opere distinte.

Grazie alla *image recognition*, è possibile analizzare e confrontare immagini di dipinti e di opere scultoree ed architettoniche. Emblematico è il caso del dipinto Sansone e Dalila, custodito alla National Gallery di Londra, la cui tradizionale attribuzione a Rubens è stata recentemente smentita da un sistema di Intelligenza Artificiale costruito per valutare l'autenticità dei dipinti¹⁰.

Come esempio concreto si può considerare il polittico "Madonna con bambino in trono tra santi", opera dei due fratelli veneti Carlo e Vittore Crivelli e custodito a Monte San Martino. La didascalia può non essere sufficiente per un allineamento, dato che si tratta di termini molto comuni e generici. Per capire in un confronto se le informazioni sono riferite alla stessa opera sarebbe utile poter confrontare le due immagini digitali dell'opera mostrate in figura 41.

¹⁰ Nel 2021 la start-up svizzera Art Recognition, grazie ad una rete neurale convoluzionale addestrata su 148 opere di Rubens, ha stabilito che al 91,78% il dipinto è un falso non attribuibile al pittore fiammingo.



Figura 41 – Immagini del polittico (dal catalogo dei B.C. e dal sito lavocedellemarche.it)

Le due immagini differiscono tra loro per scala, per direzione di inquadratura e per illuminazione. La libreria OpenCV permette di applicare l'algoritmo di *Computer Vision* SIFT (scale-invariant feature transform) per individuare nelle immagini *point-based features* rilevanti, invarianti alla scala e alle trasformazioni di rotazione e traslazione. L'algoritmo riesce ad individuare *feature points* stabili in tutte le scale possibili, grazie ad una funzione *Scale-space* che convoluzione con Kernel Gaussiana. Successivamente l'algoritmo applica una funzione DOG (*Difference of Gaussian*) per localizzare i *keypoint*, ne determina l'orientazione e, calcolando il gradiente dei punti che si trovano nei loro intorno, associa ad ogni *keypoint* un descrittore contenente gli istogrammi delle orientazioni. I descrittori sono vettori normalizzati che permettono di mettere in relazione *keypoint* in immagini diverse in base alla loro vicinanza.

Per confrontarne il contenuto delle immagini dell'opera dei Crivelli, con uno script python si trasformano le immagini in scala di grigio e si applica l'algoritmo SIFT. L'operazione di *matching* che mette in corrispondenza i punti individuati sulle due immagini del polittico è illustrata in figura 42. Analisi più sofisticate con l'ausilio di reti neurali possono restituire la probabilità che si tratti di riproduzioni della stessa opera.

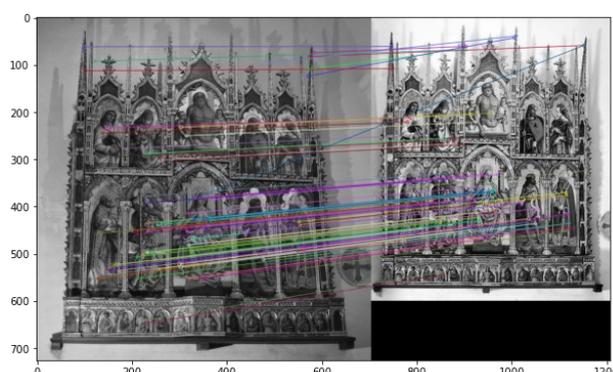


Figura 42 – Matching tra i keypoints delle due immagini del polittico

5.4 Analisi del grafo K-MAG

Grazie alla libreria networkX di python è possibile analizzare il contenuto del grafo K-MAG o di sotto-grafi ottenuti da viste o estrazioni di nodi e archi. Si utilizzano metriche comuni in teoria dei grafi, come il *grado* di un nodo (numero di archi incidenti), la *densità* (numero di connessioni esistenti rispetto al numero di connessioni potenziali), l'*eccentricità* di un nodo (distanza massima da questo a tutti gli altri nodi del grafo), che quando viene calcolata su tutti i nodi del grafo ne determina il *raggio* (minima eccentricità) e il *diametro* (massima eccentricità), evidenziando il *centro* (insieme dei nodi con eccentricità uguale al raggio) e la *periferia* (insieme dei nodi con eccentricità uguale al diametro). Altre metriche, come il *coefficiente di clustering*, misura quanto i nodi di un grafo sono interconnessi tra loro.

Se si estrae dal grafo K-MAG originale il sotto-grafo dei Musei che custodiscono opere di artisti, si ottiene il grafo diretto avente 673 nodi (in rosso sono evidenziati i Musei, in giallo le Opere, in azzurro gli Artisti) e 1.021 archi diretti, visualizzato in figura 43. La densità, data per i grafi diretti dalla formula $D = \frac{e}{n(n-1)}$, è 0,00226.

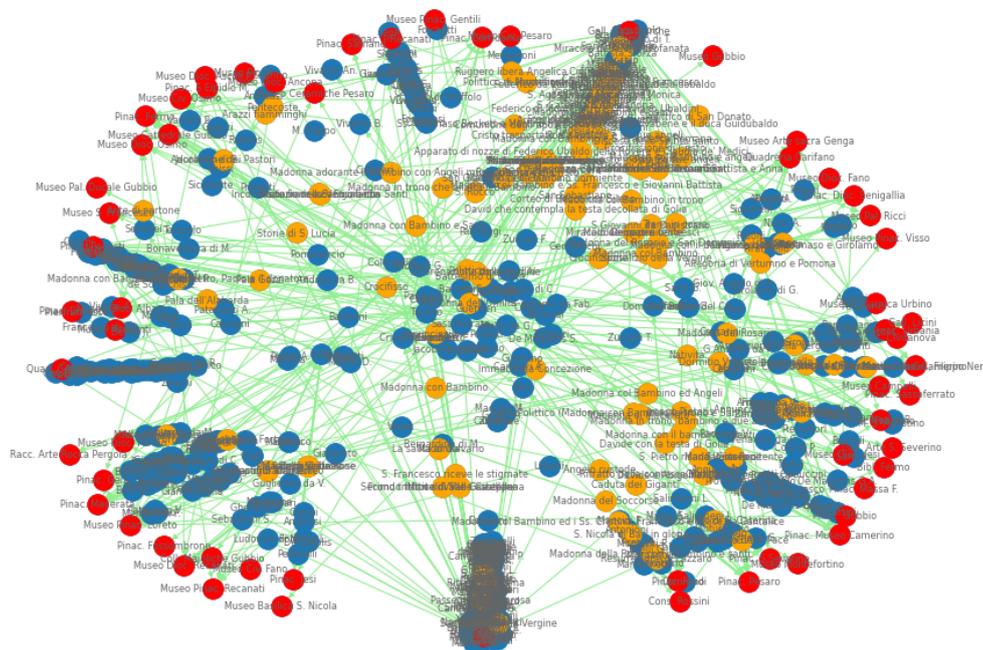


Figura 43 – Sotto-grafo dei Musei, con artisti e opere notevoli

È possibile ripetere questa analisi dopo avere effettuato il processo di integrazione dati, ottenendo un grafo formato da 11.188 nodi e 20.796 archi, con densità 0,00017.

I Musei e gli artisti con più connessioni prima e dopo l'integrazione dati in K-MAG sono indicati in tabella 15. È interessante notare che nel grafo originale erano presenti poche opere notevoli e citati gli artisti più famosi. Dopo l'integrazione con i dati dei Beni Culturali gli artisti più connessi sono artisti più recenti prevalentemente locali (Ercolani e Carbonati del Novecento, Gabrielli, Duranti e Paci dell'Ottocento).

sotto-grafo con dati originali	sotto-grafo con dati dopo l'integrazione
<p><i>Musei con più connessioni:</i> 165 Pinacoteca di Ascoli Piceno 140 Galleria Nazionale delle Marche 51 Pinacoteca Podesti 33 Quadreria Cesarini 29 Pinacoteca Fano</p>	<p><i>Musei con più connessioni:</i> 1.357 Pinacoteca di Ascoli Piceno 1.116 Biblioteca Comunale "Mozzi - Borgetti" 1.075 Museo Civico di Urbania 847 Galleria d'Arte Contemporanea O. Licini 817 Galleria Nazionale delle Marche</p>
<p><i>Artisti con più connessioni:</i> 17 Lorenzo Lotto e Vittore Crivelli 16 Federico Barocci 11 Simone Cantarini, Carlo Crivelli, Guercino, Giovanni Guerrieri e Vincenzo Pagani 10 Simone De Magistris 8 Guido Reni</p>	<p><i>Artisti con più connessioni:</i> 259 Ernesto Ercolani 198 Giulio Gabrielli 149 Fortunato Duranti 119 Antonio Carbonati 101 Emidio Paci 89 Antonio Basoli 84 Annibale Carracci</p>

Tabella 15 – Musei e Artisti con più connessioni prima e dopo l'integrazione

Estraendo il sotto-grafo relativo alla Galleria Nazionale delle Marche si ottengono i grafi diretti in figura 44, con 141 nodi e 220 archi diretti prima dell'integrazione, 796 nodi e 1.506 archi dopo l'integrazione dei dati. Gli autori che figurano con più connessioni sono attribuzioni non certe ad artisti e botteghe. Gli autori certi con più opere sono Giusto di Gand e Pedro Berruguete, entrambi chiamati alla corte da Federico di Montefeltro.

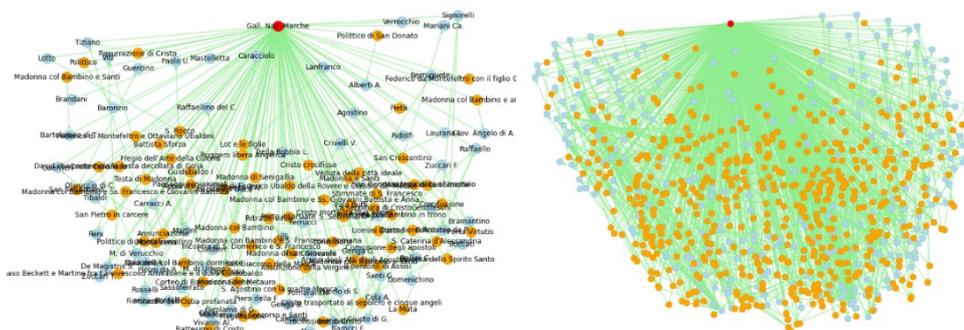


Figura 44 – Sotto-grafi della Galleria Nazionale delle Marche prima e dopo l'integrazione

Come applicazioni di analisi e utilizzo di informazioni sono state eseguite le seguenti operazioni sui dati del Knowledge Graph K-MAG:

1. Social Network degli artisti che hanno eseguito interventi e/o hanno opere contenute nello stesso sito fisico.
2. Localizzazione dei musei o delle opere su una mappa geografica e ricerca di un percorso di visita delle opere di un autore nel dominio geografico di interesse.

5.4.1 Social Network degli artisti più collegati

Per trovare gli Artisti più collegati, è stata generata una vista del grafo K-MAG. I nodi della vista sono tutti gli artisti che hanno un'opera attribuita nel dominio di interesse e gli archi collegano artisti che hanno eseguito interventi e/o hanno opere contenute nello stesso sito. Ogni arco ha un peso uguale al numero di siti contenitori condiviso dai due artisti. Il grafo ottenuto contiene 771 nodi e 21.415 archi pesati, con una densità, data dalla formula $D = \frac{2e}{n(n-1)}$, di 0,072.

Il grado massimo del grafo è 289 e la frequenza dei gradi dei nodi in scala logaritmica è illustrata in figura 45 insieme ad una rappresentazione visuale del grafo.

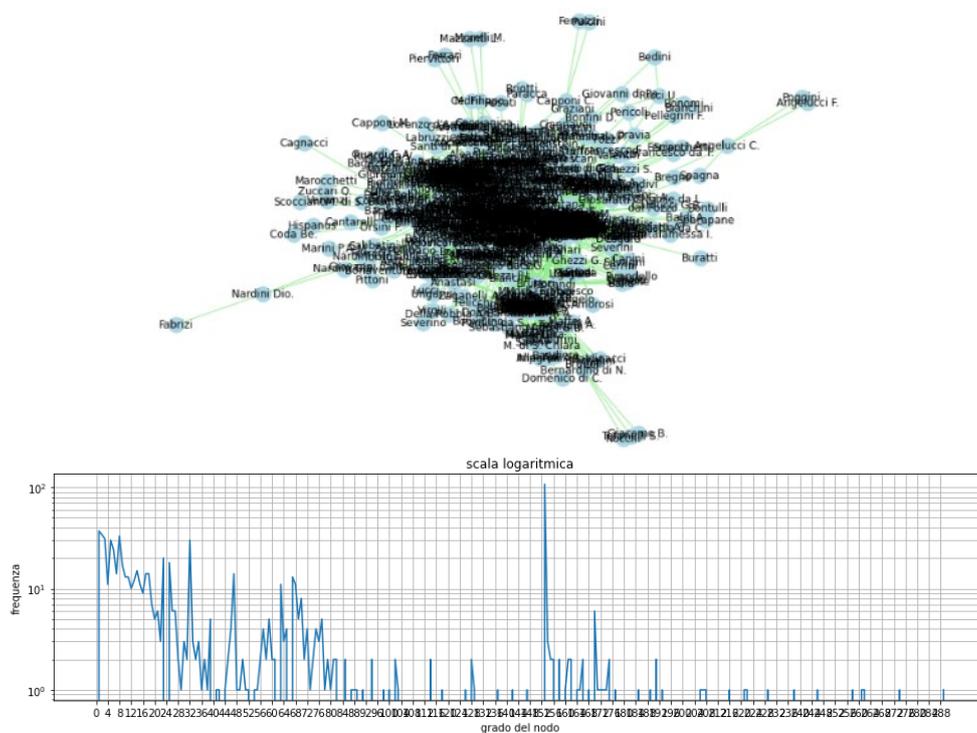


Figura 45 – Social Network degli artisti e frequenza dei gradi in scala logaritmica

Guercino è l'artista che in questa Social Network ha più connessioni pesate e non pesate con gli altri artisti. In figura 46, tra gli artisti con più connessioni risultano presenti cinque marchigiani (De Magistris, Maratta, Pagani e Sassoferrato, Zuccari), tre emiliani (Carracci, Guercino e Reni), due veneti (Crivelli e Tiziano), un toscano (Pomarancio) e un umbro-laziale (Cola)



<i>connessioni non pesate:</i>	<i>connessioni pesate:</i>
289 Guercino	340 Guercino
274 Sassoferrato	304 Simone De Magistris
262 Simone De Magistris	302 Sassoferrato
261 Vincenzo Pagani	294 Vincenzo Pagani
258 Carlo Maratta	291 Carlo Maratta
246 Tiziano	270 Tiziano
238 Guido Reni	263 Guido Reni
229 Annibale Carracci	249 Annibale Carracci e Carlo Crivelli
222 Carlo Crivelli	243 Cola dell'Amatrice e Pomarancio
221 Cola dell'Amatrice	231 Taddeo Zuccari

Figura 46 – Ritratto del Guercino e artisti con più connessioni pesate e non pesate

Calcolando l'eccentricità del grafo si determina un raggio pari a 4 e un diametro pari a 7. La figura 47 mostra in rosso il centro del grafo, composto da 164 artisti, e in giallo la periferia, composta da 9 artisti. Il baricentro del grafo, definito come l'insieme di nodi che ottimizzano la funzione obiettivo, è ridotto ad un solo elemento, che nel caso non pesato è il Guercino, nel caso pesato è il Sassoferrato. Il coefficiente di clustering medio è 0,825 per il grafo non pesato e 0,145 per il grafo con archi pesati.

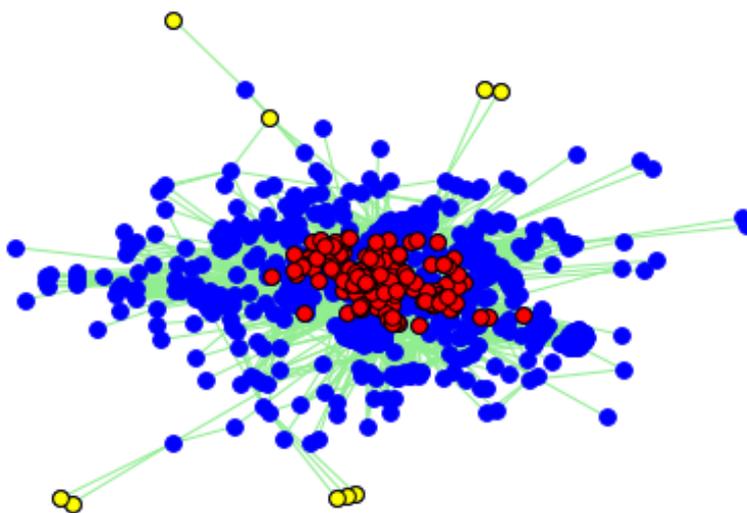


Figura 47 – Centro e periferia della Social Network degli artisti

In figura 49 sono visualizzate le Ego Network degli artisti più collegati nelle quattro macro-aree, Pomarancio (Ancona), Vincenzo Pagani (Ascoli e Fermo), Maratta (Macerata e Camerino) e Rosselli (Pesaro e Urbino con Gubbio). Sono evidenti i cluster di nodi collegati per il fatto di condividere opere all'interno dei Musei più rilevanti.

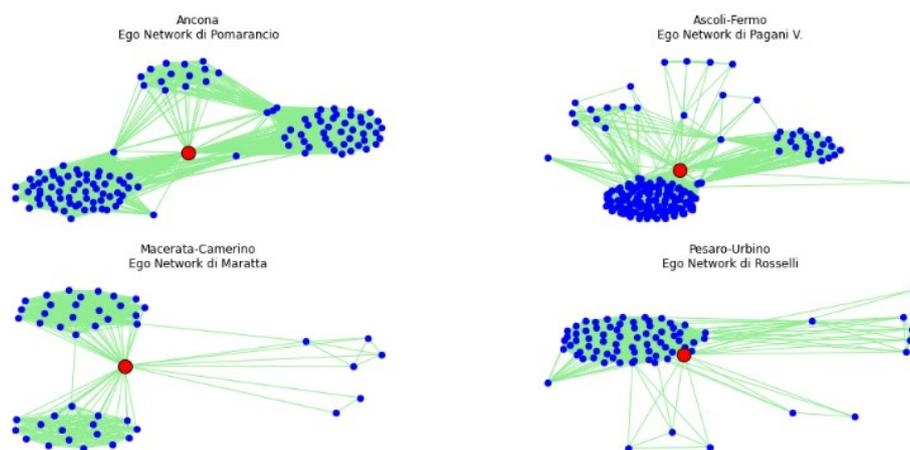


Figura 49 – Ego Network degli artisti più collegati nelle quattro macro-aree geografiche

La centralità è un concetto di Social Network Analysis che descrive l'importanza di un singolo nodo all'interno di una rete, ed è possibile definire diversi *indici di centralità* in base al significato attribuito alla parola importanza:

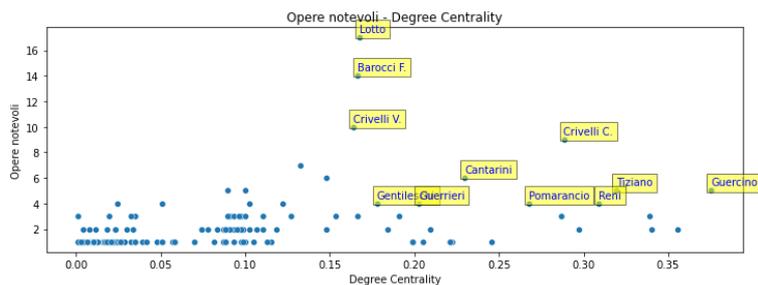
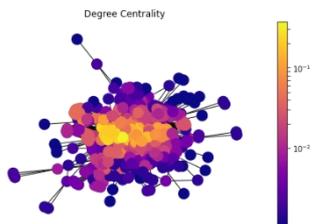
- *degree centrality* è il numero di connessioni di un nodo rispetto al numero totale di connessioni possibili;
- *closeness centrality* considera la centralità come vicinanza a tutti gli altri nodi; un nodo è più centrale se ha minore distanza media da tutti gli altri nodi;
- *betweenness centrality* definisce la capacità di un nodo di connettere tutti gli altri nodi; un nodo più centrale incrocia più percorsi minimi tra coppie di nodi rispetto al numero totale dei percorsi minimi.

Altre misure di centralità suppongono che l'importanza di un nodo sia determinata dall'importanza dei suoi nodi vicini:

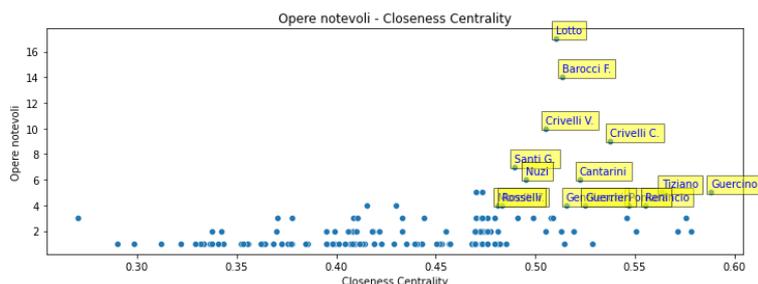
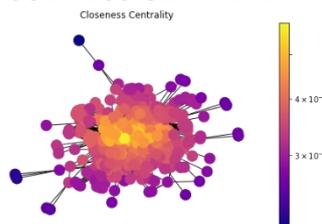
- *eigenvector centrality* calcola la centralità mediante il calcolo degli autovettori della matrice di adiacenza;
- *Page rank* deriva dall'algoritmo di Google, nato per definire l'ordine mostrato nei risultati di ricerca in base alla struttura dei collegamenti.

La figura 50 mostra la distribuzione degli indici di centralità della Social Network in correlazione con il numero di opere notevoli presenti nel grafo K-MAG.

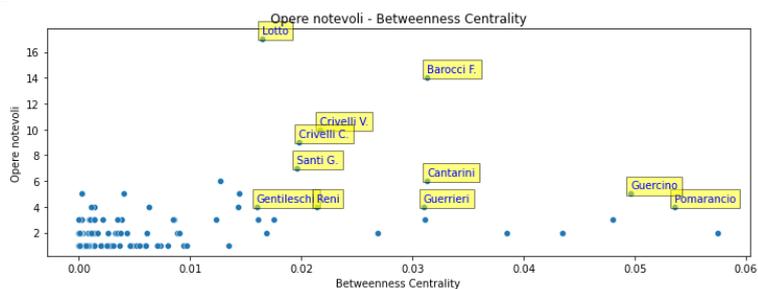
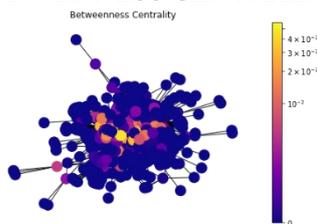
DEGREE CENTRALITY



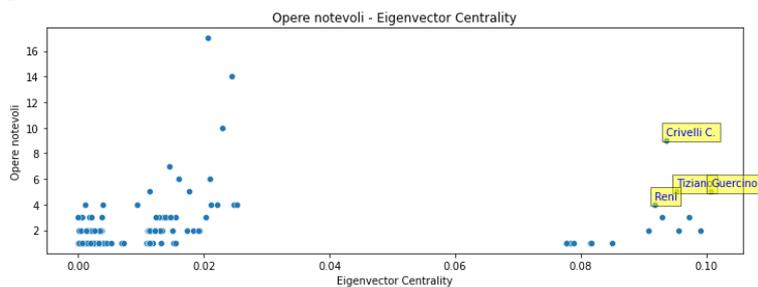
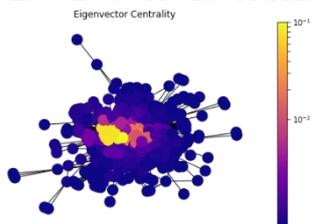
CLOSENESS CENTRALITY



BETWEENNESS CENTRALITY



EIGENVECTOR CENTRALITY



PAGE RANK

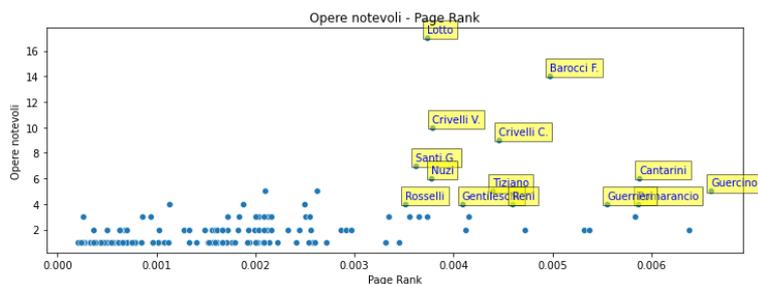
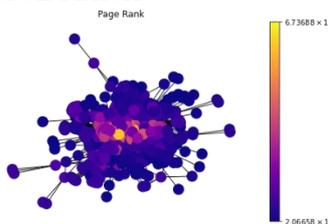


Figura 50 – Distribuzione degli indici di centralità e correlazione con il numero di opere

Si nota che per l'indice *eigenvector centrality* ci sono due insiemi di artisti nettamente separati, mentre la distribuzione negli altri indici risulta più uniforme. La tabella 17 confronta gli indici di centralità misurati sulla Social Network dell'intera regione e nelle quattro sotto-reti per ognuna delle macro-aree geografiche definite precedentemente.

REGIONE COMPLETA	ANCONA	ASCOLI-FERMO	MACERATA-CAMERINO	PESARO-URBINO-GUBBIO
<i>Degree Centrality:</i>				
0,375 Guercino	0,523 Pomarancio	0,748 Pagani V.	0,218 Maratta	0,483 Ridolfi
0,356 Sassoferrato	0,346 Bellini F.	0,674 Crespi, Morelli, De Carolis e Pettinelli	0,185 De Magistris S.	0,449 Guerrieri
0,340 De Magistris S.	0,346 Silva		0,156 Girolamo di G.	0,432 Santi G.
<i>Closeness Centrality:</i>				
0,588 Guercino	0,621 Pomarancio	0,780 Pagani V.	0,439 Crivelli C.	0,624 Ridolfi
0,579 Sassoferrato	0,577 Bellini F.	0,734 Crespi, Morelli, De Carolis e Pettinelli	0,421 Barocci F. e Lorenzo d'Al.	0,617 Guerrieri
0,575 Pagani V.	0,556 Gentileschi			0,560 Cantarini
<i>Betweenness Centrality:</i>				
0,057 De Magistris S.	0,145 Pomarancio	0,092 Pagani V.	0,180 Maratta	0,228 Biancini
0,054 Pomarancio	0,107 Bellini F.	0,072 Fontana L.	0,126 Girolamo di G.	0,170 Ridolfi
0,050 Guercino	0,084 Gentileschi	0,033 Crivelli C.	0,114 De Magistris S.	0,136 Guerrieri
<i>Eigenvector Centrality</i>				
0,101 Guercino	0,195 Calcagni	0,086 Crivelli C. e Pagani V.	0,230 Maratta	0,209 Rosselli
0,099 Sassoferrato	0,193 Della Porta		0,206 Sassoferrato	0,194 Pontelli
0,098 Maratta	0,192 Cirilli	0,085 Paci G. e Trasi	0,203 Severini	0,181 Martini
<i>Page Rank:</i>				
0,0066 Guercino	0,013 Gentileschi	0,0106 Pagani V.	0,019 De Magistris S.	0,0160 Barocci F.
0,0064 De Magistris S.	0,012 Pomarancio	0,0082 Fontana L.	0,014 Lorenzo d'Al.	0,0158 Guerrieri
0,0059 Cantarini	0,011 Bellini F.	0,0076 Paci G.	0,013 Maratta	0,0156 Ridolfi

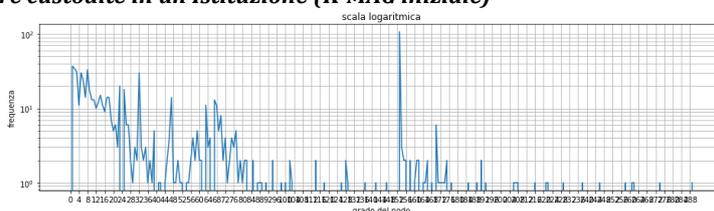
Tabella 17 – Indici di centralità in tutta la regione e nelle quattro macro-aree

Nell'analisi sono state considerate tutte le strutture e le sedi in cui gli artisti hanno eseguito interventi o sono contenute loro opere. È interessante ripetere questa analisi per le sole opere contenute nei Musei con una vista dal grafo K-MAG originale e con una vista dopo avere effettuato il processo di integrazione dati da RDF Store. In figura 51 si confronta la densità delle due reti e la frequenza dei gradi dei nodi.

Social network degli Artisti con opere custodite in un'Istituzione (K-MAG iniziale)

numero nodi: 456
numero archi: 17.393
densità del grafo: 0,168
grado massimo: 271

frequenza del grado dei nodi >
(scala logaritmica)



Social network degli Artisti con opere custodite in un'Istituzione (K-MAG dopo l'integrazione)

numero nodi: 2.899
numero archi: 317.714
densità del grafo: 0,076
grado massimo: 1.002

frequenza del grado dei nodi >
(scala logaritmica)

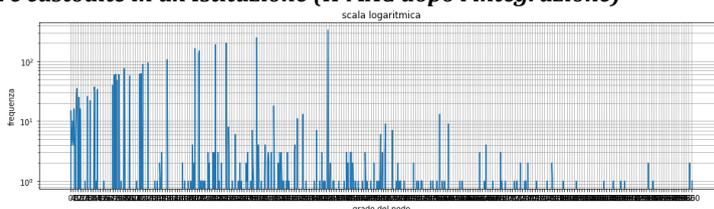


Figura 51 - Social Network artisti nei musei, prima e dopo l'integrazione dati in K-MAG

In tabella 18 si confrontano gli artisti con i più elevati indici di centralità prima e dopo l'integrazione del grafo K-MAG con i dati RDF dei Beni culturali. In figura 52 viene mostrata la correlazione tra gli indici *Degree* ed *Eigenvector Centrality* e le opere presenti in un museo. È interessante notare che artisti come l'ascolano Ernesto Ercolani hanno un grande numero di opere e meno centralità, mentre artisti come Guido Reni e Giovanni Guerrieri sono più collegati a tutti gli artisti della regione.

Artisti presenti in K-MAG con opere custodite in un Museo o Istituto Culturale

sotto-grafo con dati originali	sotto-grafo con dati dopo l'integrazione
<i>Degree Centrality:</i> 0,598 Guercino 0,596 Sassoferrato	<i>Degree Centrality:</i> 0,3458 Guido Reni 0,3444 Alessandro Gallucci e Giovanni Guerrieri
<i>Closeness Centrality:</i> 0,709 Guercino 0,707 Sassoferrato	<i>Closeness Centrality:</i> 0,5933 Guido Reni 0,5931 Giovanni Guerrieri
<i>Eigenvector Centrality:</i> 0,099 Sassoferrato 0,097 Guercino	<i>Eigenvector Centrality:</i> 0,0652 Wladimiro Tulli 0,0652 Alessandro Gallucci
<i>Page Rank:</i> 0,0086 Guercino 0,0078 Sassoferrato	<i>Page Rank:</i> 0,0018 Giovanni Guerrieri 0,0016 Guido Reni

Tabella 18 – Social Network artisti nei musei: indici di centralità

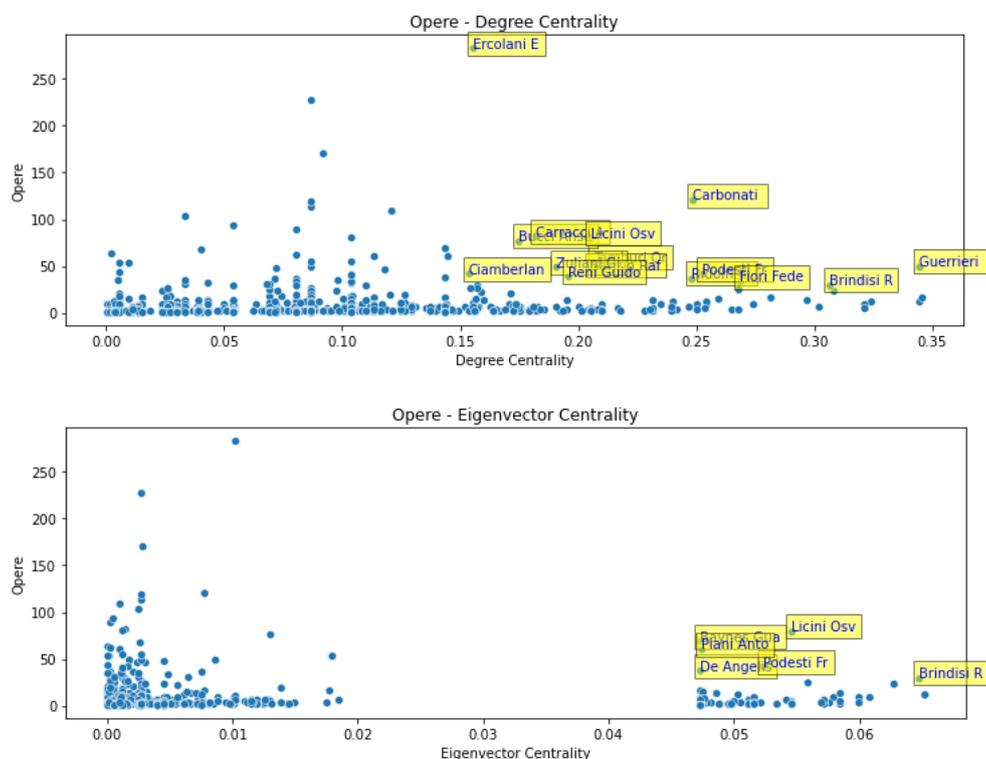


Figura 52 - Correlazione tra indici di centralità e opere nei musei dopo l'integrazione

5.4.2 Localizzazione delle opere e ricerca di un percorso di visita

La geolocalizzazione delle entità fisiche permette di costruire mappe dei Siti e delle Istituzioni Culturali che contengono Beni artistici e archeologici.

Se si estraggono dal grafo K-MAG le sedi dei Musei e si associano con le coordinate geografiche delle località, la libreria *folium* di Python permette di costruire una mappa geografica interattiva, di inserire *marker* per evidenziare i punti notevoli e *tooltip* utili e fornire informazioni. La mappa è esportabile in HTML ed è possibile ingrandire l'area visibile a diverse scale (figura 53).

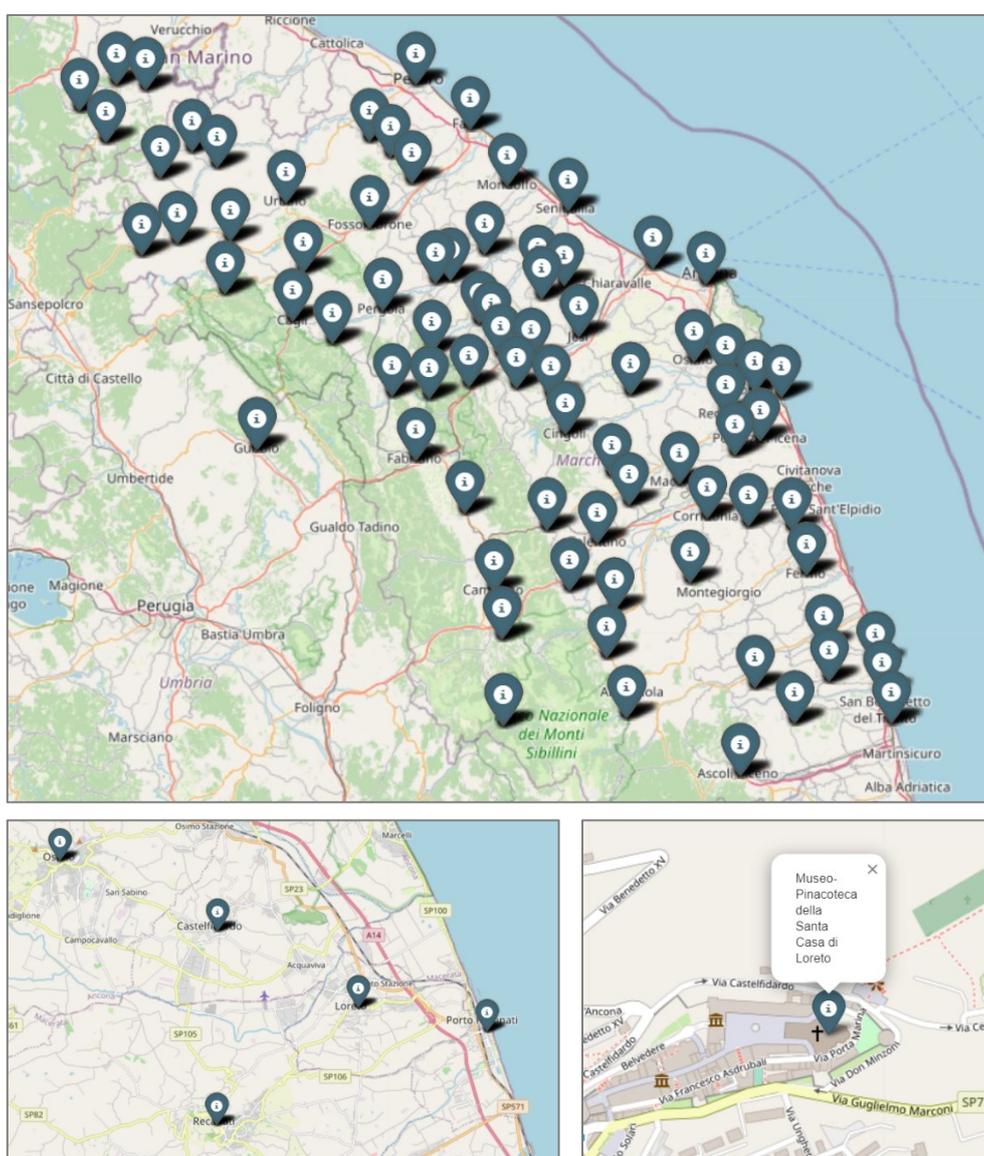


Figura 53 – Mappa interattiva dei Musei K-MAG, con possibilità di ingrandimento in scala

Per ottenere informazioni più precise sulla localizzazione dei Beni Culturali si può fare riferimento alle due ontologie utilizzate da ArCo che descrivono informazioni di tipo geografico. L'ontologia degli Indirizzi/Luoghi del Catalogo Nazionale dei Dati (CLV-AP_IT, <https://w3id.org/italia/onto/CLV/>) associa tramite il predicato `hasGeometry` una classe `Geometry`, che rappresenta la geometria di una entità di tipo spaziale. Esiste anche una ontologia specifica per i Beni Culturali che ne rappresenta la posizione (modulo `a-loc:https://w3id.org/arco/ontology/location/`). Questa ontologia con il predicato `hasCoordinates` associa entità `Geometry` ad una risorsa di tipo `Coordinates`, che rappresenta un insieme di coordinate che permettono di georeferenziare un oggetto puntuale, lineare areale (figura 54).

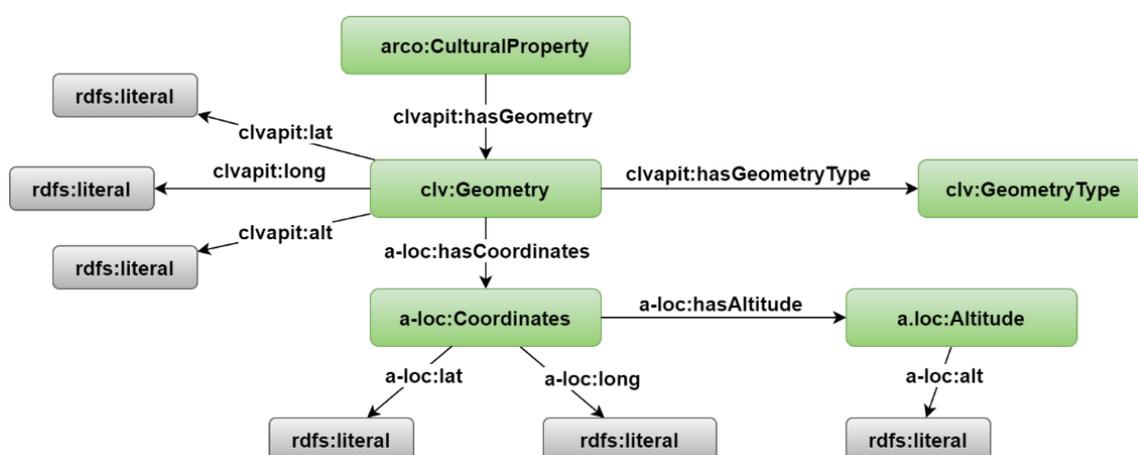


Figura 54 – Coordinate geografiche nelle ontologie Location ArCo e CLV-AP_IT

Con una query sull'endpoint SPARQL del Ministero dei Beni Culturali è possibile ottenere per le istanze delle sottoclassi della classe `CulturalProperty` le coordinate spaziali *latitudine* e *longitudine* nel sistema geografico di riferimento rispetto alla superficie terrestre (figura 55). Quindi è possibile localizzare sulla mappa la posizione delle opere d'arte per le quali sono disponibili le coordinate, e per estensione le opere d'arte custodite negli stessi musei o negli stessi edifici. In figura 56 si può vedere la localizzazione del dipinto "La Muta" di Raffaello, effettuata con uno script Python che richiama i dati delle coordinate geografiche dall'endpoint SPARQL e fissa un *marker* sulla sua posizione sulla mappa generata con la libreria `folium`. La mappa è interattiva, è possibile effettuare ingrandimenti fino alla risoluzione massima e selezionando il *marker* visualizzare un *tooltip* informativo.

```

SELECT DISTINCT ?bc, MAX(?address) AS ?indirizzo, ?lat, ?long
WHERE { ?bc rdf:type arco:HistoricOrArtisticProperty ;
  rdfs:label ?label ;
  arco:regionIdentifier "11" ;
  a-loc:hasCulturalPropertyAddress/rdfs:label ?address ;
  clvavit:hasGeometry/a-loc:hasCoordinates/a-loc:lat ?lat ;
  clvavit:hasGeometry/a-loc:hasCoordinates/a-loc:long ?long }

```

bc	indirizzo	lat	long
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100135999	"ITALIA, Marche, MC, Gagliole, GAGLIOLE"	"43.238192"	"13.067353"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100360527	"ITALIA, Marche, AN, Fabriano, via frazione cacciano"	"43.278598"	"12.842703"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100264050	"PU, Urbino, URBINO"	"43.723547"	"12.636441"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100117732	"Italia, Marche, PU, Fano"	"43.845215"	"13.017747"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100132444	"Italia, Marche, PU, Fossombrone, via Pergamino, 24"	"43.687537"	"12.798599"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100147473	"ITALIA, Marche, AN, Sassoferrato"	"43.444904"	"12.830821"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100062737-5	"Italia, Marche, MC, Corridonia"	"43.248819"	"13.508808"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100141286	"Italia, Marche, AP, Fermo"	"43.161469"	"13.718271"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100029678	"Italia, Marche, AP, Ascoli Piceno"	"42.853715"	"13.578494"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100141397-15	"Italia, Marche, AP, Fermo"	"43.161469"	"13.718271"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100115007	"Italia, Marche, MC, Sarnano, via Giacomo Leopardi"	"43.035694"	"13.300824"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100053258	"Italia, Marche, AN, Staffolo"	"43.433319"	"13.185968"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100102061	"Italia, Marche, AP, Santa Vittoria in Matenano"	"43.019516"	"13.495976"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100052576	"Italia, Marche, MC, Recanati, Via Roma"	"43.400885"	"13.552371"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100104738-1	"Italia, Marche, AP, Montegranaro"	"43.234364"	"13.634358"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100114956	"Italia, Marche, MC, Sarnano, via Giacomo Leopardi"	"43.035694"	"13.300824"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100273344	"ITALIA, Marche, PU, Urbania, Corso Vittorio Emanuele, 23"	"43.668132"	"12.523235"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100139617	"Italia, Marche, AP, Ascoli Piceno"	"42.853613"	"13.577513"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100029888	"Italia, Marche, AP, Ascoli Piceno"	"42.853715"	"13.578494"
https://w3id.org/arco/resource/HistoricOrArtisticProperty/1100033452	"PU, Urbino"	"43.723409"	"12.636973"

Figura 55 – Query SPARQL per ottenere le coordinate geografiche di un Bene Culturale

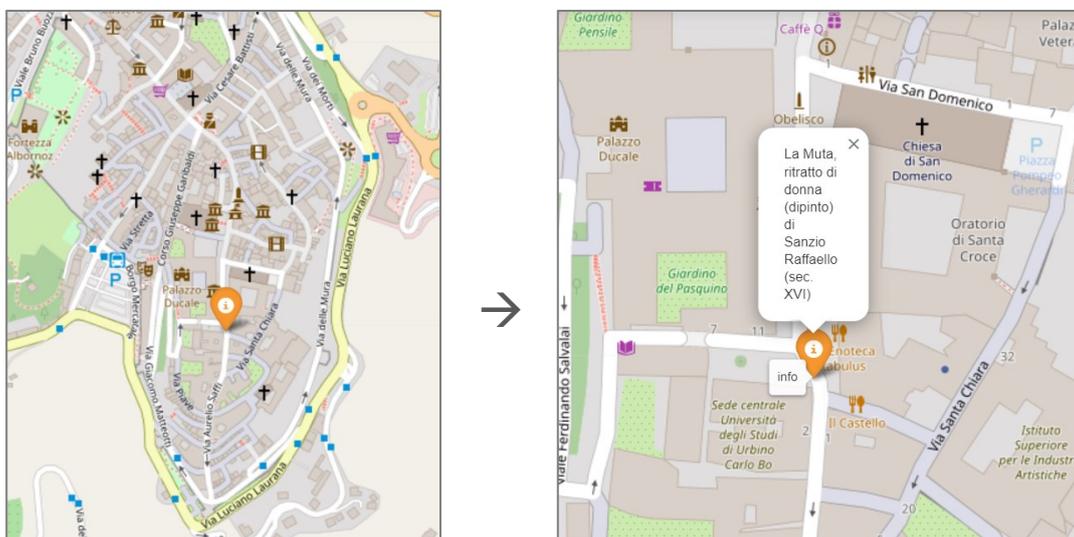


Figura 56 – Localizzazione sulla mappa del dipinto "La Muta" di Raffaello

Il passo successivo più interessante è scoprire quale sia un percorso ottimizzato per vedere le diverse opere di un determinato autore attraverso il dominio geografico di interesse.

Consideriamo il caso più semplice del percorso più breve tra due punti, supponendo di voler visitare opere di Carlo Crivelli nelle Pinacoteche di Macerata e Ascoli Piceno. Partendo dalle coordinate dei due luoghi è possibile richiamare le API di OpenStreetMap, che restituiscono un file JSON contenente l'insieme delle coordinate geografiche dei punti del cammino che congiunge i due punti. Questo insieme di punti può essere visualizzato come un percorso sulla mappa utilizzando la libreria plotly di Python (figura 57).

Partenza: "Macerata", latitudine: "43,299167", longitudine: "13,452222"
 Destinazione: "Ascoli Piceno", latitudine: "42,853333", longitudine: "13,578056"

Lista delle coordinate dei nodi attraversati nel percorso:

	Node	coordinates	lat	long
	0	(43.2974044, 13.5209701)	43.297404	13.520970
	1	(43.2968970, 13.5217029)	43.296897	13.521703
	2	(43.2960533, 13.5223189)	43.296053	13.522319
	3	(43.2954117, 13.5225615)	43.295412	13.522562
	4	(43.2949241, 13.5228426)	43.294924	13.522843

	493	(42.8537060, 13.5786916)	42.853706	13.578692
	494	(42.8543608, 13.5786047)	42.854361	13.578605
	495	(42.8545454, 13.5778727)	42.854545	13.577873
	496	(42.8541281, 13.5778983)	42.854128	13.577898
	497	(42.8537765, 13.5779575)	42.853777	13.577958

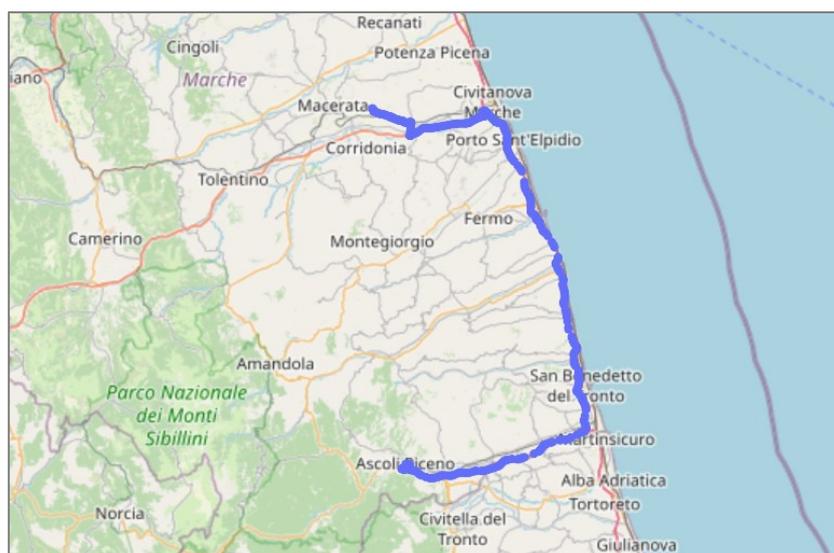


Figura 57 – Percorso migliore tra due luoghi calcolato richiamando le API di OpenStreetMap

Questo problema può anche essere risolto all'interno del Graph DB Neo4j, aggiungendo un nuovo tipo di arco che collega i nodi `Localita`, con etichetta `:ADIACENTE_A` e le proprietà `distanza` e `tempo_di_percorrenza`, che devono essere preventivamente calcolati o scaricati da fonti opportune. Richiamando l'algoritmo `shortestPath` è possibile trovare il cammino minimo tra due nodi, valutando la funzione di costo (in questo caso il tempo di percorrenza minimo).

Più complesso è il problema di scegliere il percorso ottimale tra un insieme di punti, richiedendo di visitare i luoghi una sola volta. Si tratta del noto "problema del commesso viaggiatore", indicato con il suo acronimo inglese TSP (Traveling Salesman Problem). È stato dimostrato che TSP è un problema NP-hard, sia nella versione completa che in versioni ridotte, pertanto non esistono algoritmi efficienti per la sua risoluzione. Un possibile metodo di risoluzione è l'enumerazione totale dei percorsi, ovvero si elaborano tutti i possibili cammini sul grafo e si sceglie quello ottimale rispetto alla funzione di costo. Ma questa operazione risulta già molto complessa per reti con mille nodi.

6 Conclusioni e prospettive future

Il tema dell'integrazione dei dati e del confronto tra i diversi modelli a grafo è un tema molto interessante e, considerata l'importanza dei Knowledge Graph e dei sistemi di rappresentazione della conoscenza, è lecito aspettarsi nel futuro prossimo studi e ricerche approfondite in questo campo. Storicamente [3] la rappresentazione della conoscenza ha radici lontane e l'uso di diagrammi per facilitare il ragionamento umano risale ai tempi di Aristotele (IV secolo a.C.) e percorre i secoli fino ai nostri tempi. Gli studi a fine ottocento e l'avvento dell'informatica digitale hanno portato allo sviluppo delle reti semantiche formali, sulle quali nel 1979 Sowa, alla voce "Reti semantiche" dell'Enciclopedia delle scienze cognitive, esprime questa considerazione:

"Woods (1975) e McDermott (1976) osservarono che le reti semantiche stesse non hanno una semantica ben definita. Il calcolo dei predicati standard ha una semantica teorica del modello definita con precisione ed è adeguato per descrivere teorie matematiche con un insieme chiuso di assiomi. Ma il mondo reale è disordinato, esplorato in modo incompleto e pieno di sorprese inaspettate."

I lavori sui grafi concettuali e sulle reti semantiche sono i predecessori della logica descrittiva, che intende costruire sistemi di ragionamento pratico per le logiche decidibili che costituiscono le basi dello standard OWL. Dopo l'annuncio di Google del 2012 sono emerse diverse definizioni accademiche dei Knowledge Graph, alcune di tipo intensionale, altre di tipo estensionale. Il mondo aziendale è molto interessato [27] all'argomento: "Knowledge Graph e strutture simili forniscono un substrato condiviso di conoscenza all'interno di un'organizzazione, consentendo a diversi prodotti e applicazioni di utilizzare un vocabolario simile e di riutilizzare definizioni e descrizioni create da altri. Inoltre forniscono una rappresentazione formale compatta che gli sviluppatori possono utilizzare per dedurre nuovi fatti e sviluppare conoscenze".

Il presente lavoro ha illustrato la definizione dei sistemi di rappresentazione della conoscenza sottoforma di grafo, evidenziando pregi, difetti e differenze dei due principali modelli, Property graph e RDF Graph, confrontando i principali linguaggi di interrogazione sui Graph DB, SPARQL e Cypher, in un dominio concreto che comprende l'insieme dei Beni Culturali della regione storico-geografica marchigiana.

Il problema principale emerso è a livello di interpretazione semantica e riguarda tutti gli algoritmi che si occupano di conoscenza. È collegato al rapporto tra quanta conoscenza dipende dai dati e quanta conoscenza dipende dall'interpretazione che diamo quando leggiamo i dati, grazie a quanto abbiamo imparato precedentemente.

La costruzione di una ontologia, con un sistema di metadati che vuole dare significato semantico governando ogni relazione, rischia di rendere il sistema molto farraginoso e non risolve il problema dell'identità, basata su URI molteplici e non sempre condivise.

La conoscenza in forma di Property Graph è più semplice da costruire e gestire, ma chi guarda il grafo è costantemente posto di fronte al problema dell'interpretazione delle informazioni, per la quale è richiesto un minimo di esperienza del dominio.

Nel caso dei Beni Culturali il problema si amplifica a causa della molteplicità dei soggetti che detengono e certificano le informazioni, con una dimensione temporale delle interpretazioni delle opere d'arte e delle attribuzioni ad artisti o a botteghe collegate, con una indeterminazione dei tempi e dei luoghi, con una difficoltà di classificare in modo univoco le opere e i siti, le categorie.

Per questo il problema di integrare basi di conoscenza tra loro eterogenee ricorda il problema della traduzione di testi, che dopo aver percorso la strada delle regole grammaticali si è rivolta a metodi fondati sull'ingestione di quantità enorme di dati. Le tecniche di *machine learning* cercano di dare alle macchine con metodi statistici quella che noi chiamiamo intuizione, ovvero la capacità di capire un testo tra le righe, di scoprire similarità e differenze tra parole e testi, di effettuare classificazioni e clustering.

Prospettive interessanti sono l'estrazione di informazioni dalle descrizioni testuali presenti nel Catalogo dei Beni Culturali in formato RDF e il confronto delle immagini relative ai Beni Culturali descritti. L'estrazione di parole chiave e di informazioni semantiche dai testi risulta utile anche per ricostruire la storia di un Bene Culturale dalla sua realizzazione in poi, con la successione temporale degli interventi e dei luoghi in cui è stato collocato. Completando questi metodi con l'analisi delle immagini eseguita con strumenti avanzati di *deep learning*, si potrà integrare il Knowledge Graph con tutte le informazioni utili per estendere progressivamente la conoscenza oltre gli attuali confini.

7 Ringraziamenti

Ringrazio in primo luogo i professori Claudia Diamantini ed Emanuele Storti, che mi hanno dato la possibilità di svolgere questo elaborato e che mi hanno seguito e supportato per il lavoro di tesi. Insieme a loro ringrazio tutti i professori e gli assistenti incontrati in questi anni, seguendo le lezioni e sostenendo gli esami: Adriano Mancini, Aldo Dragoni, Alessandro Calamai, Alessandro Cucchiarelli, Alessandro Freddi, Alex Mircoli, Chiara De Fabritiis, Claudia Diamantini, Domenico Potena, Domenico Ursino, Donato Iacobucci, Emanuele Balloni, Emanuele Storti, Evghenia Sakellariadi, Fabrizio Marinelli, Ferdinando Pezzella, Francesca Alessio, Francesco Pagliarecci, Francesco Spegni, Gianluca Ippoliti, Gianni Albertini, Giuseppa Ribighini, Giuseppe Conte, Giuseppe Orlando, Luca Spalazzi, Luca Virgili, Lucio Demeio, Marco Farina, Matteo Franca, Michele Marchetti, Neculai Teleman, Ornella Pisacane, Paolo Crippa, Paolo Mengucci, Primo Zingaretti e Simone Fiori. Ognuno di loro mi ha trasmesso, oltre alla conoscenza delle materie di cui sono custodi per le nuove generazioni, una personale miscela di entusiasmo, rigore, passione, esperienza e curiosità.

Ringrazio tutte le ragazze e i ragazzi che in questi anni sono stati compagni di corso, troppo numerose e numerosi da elencare. Sono nodi del grafo della mia esistenza, con archi indelebili e un posto nel mio cuore. Non ho mai avvertito la distanza generazionale, ho imparato ad apprezzarne la capacità di affrontare gli ostacoli e di affrontare la vita, gli studi e il lavoro, con il coraggio di chi non ha paura di volare. Un grazie di cuore a tutti i colleghi e gli amici che mi hanno aiutato ad affrontare questo viaggio. Da Pesaro e da Ancona, da tante regioni d'Italia e all'Estero ho avvertito la stima e l'affetto con cui hanno condiviso le mie fatiche e le mie gioie. Un grazie ad Ave, sorella che mi ha instillato fin da piccolo la passione per qualunque entità che abbia relazioni qualitative e quantitative.

Un grazie infinito a Michela, Erica e Luca. Non è facile convivere per dieci anni con un marito e padre che decide di partire per un lungo viaggio sottraendo energie e tempo alla famiglia, richiedendo a tutti qualche sacrificio nei giorni dedicati al riposo. Se sono giunto al termine di questa Odissea è soprattutto merito loro.

“Fatti non foste a viver come bruti, ma per seguir virtute e canoscenza” (Dante)

8 Riferimenti bibliografici e sitografici

- [1] Alberto Facchini: *Algebra e Matematica discreta*, Padova 2000
- [2] Amit Singhal: *Introducing the Knowledge Graph: things, not strings*. Google Blog. May 16, 2012
- [3] A. Hogan, E. Blomqvist, M. Cochez, C. D'Amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier, et al.: *Knowledge graphs*, Synth. Lect. Data, Semant. Knowl., 2021
- [4] Claudia Diamantini, Domenico Potena, Emanuele Storti: *A semantic data lake model for analytic query-driven discovery*. The 23rd International Conference on Information Integration and Web Intelligence, 2021.
- [5] Renzo Angles. *The Property Graph Database Model*. In Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, vol. 2100, 2018
- [6] Keith D. Foote: *Property Graphs vs. Knowledge Graphs*, 2022, <https://www.dataversity.net/property-graphs-vs-knowledge-graphs/>
- [7] Vishnu Vettrivel: *Knowledge Graphs: RDF or Property Graphs, Which One Should You Pick?*, 2022, <https://www.wisecube.ai/blog/knowledge-graphs-rdf-or-property-graphs-which-one-should-you-pick/>
- [8] Ministero dei Beni Culturali: *Catalogo Generale dei Beni Culturali*, <https://catalogo.beniculturali.it/>
- [9] A.J.A. Donkers, D. Yang, N. Baken: *Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs*, 2020, <https://ceur-ws.org/Vol-2636/02paper.pdf>
- [10] Irene Polikoff: *Knowledge Graphs vs. Property Graphs – Part I* 2020, <https://tdan.com/knowledge-graphs-vs-property-graphs-part-1/27140>
- [11] Giuseppe Futia: *Knowledge Graph Perspectives: building bridges from RDF to LPG*, 2021, https://graphaware.com/hume/2021/09/23/kg_perspectives_RDF_to_LPG.html
- [12] Ministero dei Beni Culturali: *MiC Linked Open Data*, <https://dati.cultura.gov.it/>
- [13] Micheal Debellis: *Semantic Web vs. Property Graphs*, 2022, <https://www.michaeldebellis.com/post/owlvspropgraphs>
- [14] Francesco Battistelli: *Identità diacronica e modelli ontologici di rappresentazione di entità geografiche e amministrative*, tesi di Laurea Triennale Ing. Inf., Univpm, 2019

- [15] *TCI: Guide Rosse Touring Club delle Regioni Italiane*. Edizione Biblioteca di La Repubblica, 2005
- [16] *Italia, Guida Artistica*. Electa, 1992
- [17] *Neo4J Graph*, <https://neo4j.com/>
- [18] SIRPAC: *Catalogo dei Beni Culturali della Regione Marche*, <https://www.regione.marche.it/Regione-Utile/Cultura/Catalogo-beni-culturali>
- [19] *Sparql 1.1 query language*, <https://www.w3.org/TR/sparql11-query/>
- [20] Marcin Frackiewicz: *A Deep Dive into Graph Query Languages: Cypher, Gremlin, and SPARQL*, 2023, <https://ts2.space/en/a-deep-dive-into-graph-query-languages-cypher-gremlin-and-sparql/>
- [21] *Cypher*, <https://neo4j.com/developer/cypher/>
- [22] Renzo Angles, Harsh Thakkar, Dominik Tomaszuk: *RDF and Property Graphs Interoperability: Status and Issues*. In Proceedings of the 13th Alberto Mendelzon International Workshop on Foundations of Data Management, vol. 2369, 2019
- [23] Ezequiel José Veloso Ferreira Moreira, José Carlos Ramalho: *SPARQLing Neo4J*, 9th Symposium on Languages, Applications and Technologies (SLATE 2020).
- [24] L. Nie, S. Cao, J. Shi, J. Sun, Q. Tian, L. Hou, J. Li, J. Zhai: *GraphQ IR: Unifying the Semantic Parsing of Graph Query Languages with one Intermediate Representation*, 2022
- [25] Z. Zhao, X. Ge, Z. Shen, C. Hu, H. Wang: *S2CTrans: Building a Bridge from SPARQL to Cypher*. DEXA 2023: 424-430
- [26] Xiang Zhao, Weixin Zeng, Jiuyang Tang: *Entity Alignment: Concepts, Recent Advances and Novel Approaches* (Big Data Management), Springer 2023
- [27] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor. *Industry-scale Knowledge Graphs: Lessons and Challenges*. ACM Queue 17(2). 2019.

9 Indici

9.1 Indice delle figure

Figura 1 – Grafo e multigrafo diretto con gli insiemi dei nodi e degli archi	5
Figura 2 – Esempio di grafo diretto con archi etichettati	8
Figura 3 – RDF triple	9
Figura 4 – Esempio di RDF Graph con l’entità che rappresenta la Mole Vanvitelliana.....	10
Figura 5 – Esempio di Property Graph con l’entità che rappresenta la Mole Vanvitelliana.....	12
Figura 6 – Logo del Knowledge Graph K-MAG.....	17
Figura 7 – Delegazioni dello Stato Pontificio e Province attuali della Regione Marche.....	18
Figura 8 – Il modello Entity-Relationship di K-MAG	19
Figura 9 – Nodi e archi del grafo corrispondente al modello E-R.....	19
Figura 10 – Visualizzazione in Neo4j Browser del sotto-grafo relativo alla Mole Vanvitelliana	21
Figura 11 – Il grafo K-MAG completo	22
Figura 12 – Catalogo dei Beni Culturali nel sito della Regione Marche	23
Figura 13 – Home page del Catalogo Generale dei Beni Culturali	23
Figura 14 – Il Portale LOD dei Beni Culturali	24
Figura 15 – La scheda del Catalogo dei Beni Culturali e i corrispondenti dati in formato RDF	24
Figura 16 – Esempio di Graph pattern da mappare sul Data Graph di destra.....	25
Figura 17 – Risultati della valutazione del pattern in figura 16 con le diverse semantiche.....	26
Figura 18 – Richiesta su end point SPARQL del Ministero dei Beni Culturali e relativa risposta	28
Figura 19 - Risultato di una query su Neo4J Browser in modalità grafo, tabella, testo e codice	29
Figura 20 – Visualizzazione del cammino minimo tra due nodi in Neo4J.....	29
Figura 21 – RDF Graph dei Beni Culturali con le risorse utili per rispondere alle query	31
Figura 22 – Property Graph K-MAG con nodi, archi e proprietà utili per rispondere alle query	31
Figura 23 – Query Cypher per ottenere l’elenco dei Musei.....	32
Figura 24 – Query SPARQL per ottenere l’elenco dei Musei.....	33
Figura 25 – Query Cypher per ottenere l’elenco dei dipinti	33
Figura 26 – Query SPARQL per ottenere l’elenco dei dipinti.....	34
Figura 27 – Esempio di Entity Alignment tra due parti di Knowledge Graph.	39
Figura 28 – Workflow del processo di integrazione K-MAG	42
Figura 29 – RDF Graph dei Beni Culturali con le risorse utili per l’allineamento	43
Figura 30 – Entità K-MAG allineate con URI in valore assoluto e percentuale.....	45
Figura 31 – Lunghezza in parole e frequenza dei termini nel sito dei Beni Culturali	47
Figura 32 – Confronto tra allineamento con uguaglianza e con similarità	49
Figura 33 – Nodo del K-MAG relativo alla Mole Vanvitelliana con i valori multiproprietà URI	50
Figura 34 – Il dipinto “La Muta” di Raffaello e il suo contenitore, il Palazzo Ducale di Urbino.....	52
Figura 35 – Estrazione dell’elenco delle opere custodite nei Musei.....	54

Figura 36 – Query SPARQL e risultato con l’elenco dei Musei per numero di beni archeologici.....	55
Figura 37 – Query SPARQL e risultato con l’elenco dei Musei per numero di beni artistici	55
Figura 38 – Codice SPARQL per ottenere le opere di un Museo Archeologico	56
Figura 39 – Codici SPARQL per ottenere l’elenco dei beni artistici con gli autori	57
Figura 40 – Terza macro-fase nel workflow del processo di integrazione K-MAG.....	61
Figura 41 – Immagini del polittico (dal catalogo dei B.C. e dal sito lavocedelle Marche.it).....	63
Figura 42 – Matching tra i keypoints delle due immagini del polittico	63
Figura 43 – Sotto-grafo dei Musei, con artisti e opere notevoli	64
Figura 44 – Sotto-grafi della Galleria Nazionale delle Marche prima e dopo l’integrazione	65
Figura 45 – Social Network degli artisti e frequenza dei gradi in scala logaritmica	66
Figura 46 – Ritratto del Guercino e artisti con più connessioni pesate e non pesate	67
Figura 47 – Centro e periferia della Social Network degli artisti	67
Figura 48 – Ego Network del Guercino.....	68
Figura 49 – Ego Network degli artisti più collegati nelle quattro macro-aree geografiche	69
Figura 50 – Distribuzione degli indici di centralità e correlazione con il numero di opere.....	70
Figura 51 - Social Network artisti nei musei, prima e dopo l'integrazione dati in K-MAG	71
Figura 52 - Correlazione tra indici di centralità e opere nei musei dopo l'integrazione	72
Figura 53 – Mappa interattiva dei Musei K-MAG, con possibilità di ingrandimento in scala	73
Figura 54 – Coordinate geografiche nelle ontologie Location ArCo e CLV-AP_IT	74
Figura 55 – Query SPARQL per ottenere le coordinate geografiche di un Bene Culturale	75
Figura 56 – Localizzazione sulla mappa del dipinto "La Muta" di Raffaello.....	75
Figura 57 – Percorso migliore tra due luoghi calcolato richiamando le API di OpenStreetMap.....	76

9.2 Indice delle tabelle

Tabella 1 – Confronto tra stringhe per l'allineamento dei dati.....	40
Tabella 2 – Numerosità di entità nodo in K-MAG e risorse collegabili per classe in RDF Store	41
Tabella 3 – Tempi di elaborazione della fase di allineamento tra nodi K-MAG e URI	44
Tabella 4 – Risultati dell’allineamento tra nodi K-MAG e URI	45
Tabella 5 – Risultati in base alla similarità dell’allineamento tra i nodi Istituzione e URI	48
Tabella 6 – Tempi di elaborazione e risultati dell’allineamento con funzione di similarità	49
Tabella 7 – Tempi di elaborazione dei confronti tra proprietà K-MAG e dati RDF	51
Tabella 8 – Valori medi della funzione di similarità nei confronti tra le singole proprietà	52
Tabella 9 – Confronto delle proprietà relative al dipinto “La Muta”	53
Tabella 10 – Confronto delle proprietà relative all’artista Raffaello Sanzio	53
Tabella 11 – URI associati agli Istituti culturali selezionati	58
Tabella 12 – Collegamenti tra Artisti e Istituti in K-MAG allineati con URI da RDF Store MiC	58
Tabella 13 – Collegamenti tra Opere e Istituti in K-MAG allineati con URI da RDF Store MiC.....	59
Tabella 14 – Confronto per similarità con soglia minima 0,500 tra Istituzioni K-MAG e URI	60
Tabella 15 – Musei e Artisti con più connessioni prima e dopo l’integrazione.....	65
Tabella 16 - Artisti con più connessioni pesate nelle quattro macro-aree geografiche.....	68
Tabella 17 – Indici di centralità in tutta la regione e nelle quattro macro-aree	71
Tabella 18 – Social Network artisti nei musei: indici di centralità.....	72