ARTICLE TITLE

FABRIZIO COMINETTI, DAVIDE ABETE, RUBEN AGAZZI

CONTENTS

ABSTRACT

Il progetto realizzato consiste nella creazione di una base di dati a grafo contenente le relazioni fra i vari prodotti Marvel. All'interno del database sono presenti relazioni come ad esempio fra personaggi e fumetti, personaggi e personaggi, etc...

1 INTRODUZIONE

Lo scopo del progetto consiste nella realizzazione di un database a grafo riguardante vari prodotti Marvel, fra cui personaggi, film, serie tv e fumetti. L'obiettivo del progetto è di ottenere una sorta di "rete sociale" di super eroi o personaggi marvel, i quali sono collegati ai rispettivi fumetti, film , serie tv, o rispettivi collegamenti interpersonali fra personaggi e personaggi.

2 FONTI DATI

Per l'ottenimento dei dati sono stati utilizzati due metodi diversi: l'utilizzo di una Web API e Web Scraping.

2.1 Web Scraping

Per la parte riguardante il Web Scraping come sorgente dati è stata utilizzata la Marvel Cinematic Universe Wiki.

Lo scraping è stato effettuato eseguendo un notebook python. I passi eseguiti dal notebook per completare lo scraping sono:

1. Ottenimento dalla pagina relativa a tutti i personaggi della wiki i nomi dei personaggi con i relativi link alla pagina personale.

2. per ogni personaggio aprire la pagina personale e ottenere sempre tramite scraping le informazioni rilevanti, come ad esempio: biografia, lista di film in cui è presente il personaggio, lista di serie tv in cui è presente il personaggio e relazioni con altri personaggi.

3. Per ogni film trovato nelle pagine dei personaggi aprire la pagina relativa al film e ottenere tramite scraping le informazioni del film come ad esempio la trama, i registi, scrittori, compositori, incasso, data di uscita e durata del film.

4. Per ogni serie trovata nelle pagine dei personaggi aprire la pagina relativa alla serie e ottenere tramite scraping le informazioni della serie in questione, come ad esempio la trama, i registi, i produttori e i compositori

5. Salvataggio temporaneo all'interno di file csv di film, serie tv e personaggi per essere processati in seguito.

2.2 Web Api

Per l'ottenimento dei dati tramite web API è stata utilizzato il servizio formito dalla Marvel che mette a disposizione una sua Web Api per ottenere i dati relativi a personaggi e fumetti. L'ottenimento dei dati è avvenuto tramite esecuzione di un notebook python. L'esecuzione consiste nei seguenti passi:

1. Ottenimento dei dati dei personaggi in formato JSON tramite chiamata al relativo endpoint della web api a gruppi di 100 personaggi, in quanto è il limite imposto dagli sviluppatori del servizio.
2. Salvataggio su file CSV dei dati relativi ai personaggi ottenuti, in modo da eseguire successive manipolazioni facilmente tramite la libreria *pandas*, prima di inserire i dati nel database.
3. Ottenimento dei dati dei fumetti in formato JSON tramite chiamata al relativo endpoint della web api a gruppi di 100 fumetti, in quanto è il limite imposto dagli sviluppatori del servizio.
4. Salvataggio su file CSV dei dati relativi ai personaggi ottenuti , in modo da eseguire successive manipolazioni facilmente tramite la libreria *pandas*, prima di inserire i dati nel database..

3 DATA EXPLORATION

La fase di data exploration è stata realizzata sempre utilizzando python con alcune librerie grafiche per la creazione di visualizzazioni.

3.1 Data Quality

Le principali problematiche riscontrate nella qualità dei dati sono state riscontrate nelle biografie dei personaggi e nelle relazioni tra personaggi ottenute tramite web scraping;

3.1.1 Data Quality e Cleaning Biografia

Per quanto riguarda la qualità dei dati delle biografie le principali problematiche sono:

1. Il notebook salva i vari paragrafi delle biografie come una lista di stringhe, quindi prima di tutto la biografia viene trasformata in una stringa, concatenando le varie stringhe presenti nella lista e inserendo un carattere newLine fra un elemento e l'altro.

2. Il notebook inoltre salvava alcuni paragrafi della biografia più volte, per questo prima di concatenare la stringa viene effettuato un controllo per vedere se la porzione di stringa è già stata aggiunta alla stringa finale

3. Infine viene fatto un escaping dei caratteri speciali, come ad esempio il carattere " o ', in modo da non avere problemi nell'inserimento nel database dei dati.

3.1.2 Data Quality e Cleaning Relazioni

Per quanto riguarda la qualità dei dati delle relazioni interpersonali ottenute tramite web scraping la principale problematica consiste nell'assenza in alcune relazioni del nome del personaggio interessato, ad esempio ci sono relazioni che hanno come nome personaggio "Mother", "Father" e così via. Per risolvere questa problematica il notebook esegue le seguenti operazioni:

1. Per ogni personaggio vengono recuperate le relative relazioni.

2. Per ogni relazione ottenuta viene effettuato un controllo sul nome del personaggio della relazione, se il nome non è presente all'interno della lista di personaggi la relazione viene scartata

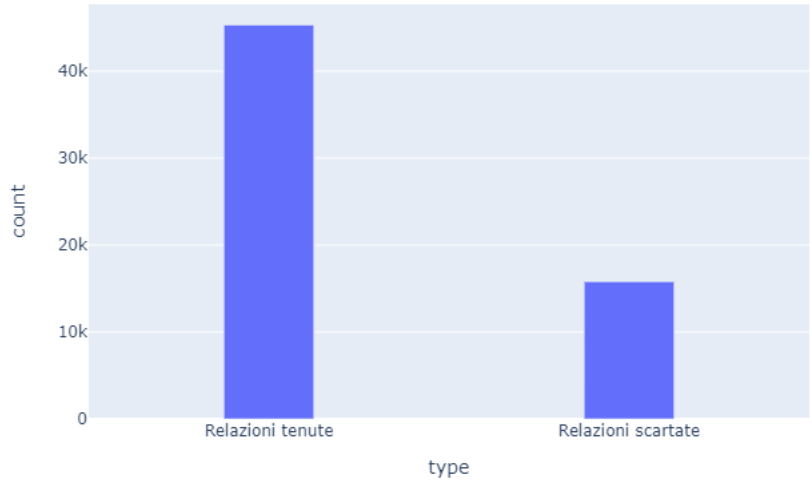

Figure 1: Grafico relativo alle relazioni tenute rispetto a quelle scartate

3.2 Completezza

Sono state effettuate altre analisi riguardanti la completezza dei dati: in particolare al riguardo di attributi dei dati ottenuti.

3.2.1 Dati personaggi

Le analisi di completezza effettuate sui personaggi consistono nel confronto fra i dati ottenuti dalla web api e quelli ottenuti tramite web scraping. In particolare si può riscontrare una corrispondenza di 323 personaggi della web api con quelli ottenuti tramite web scraping, al netto dei 1559 personaggi della web api.

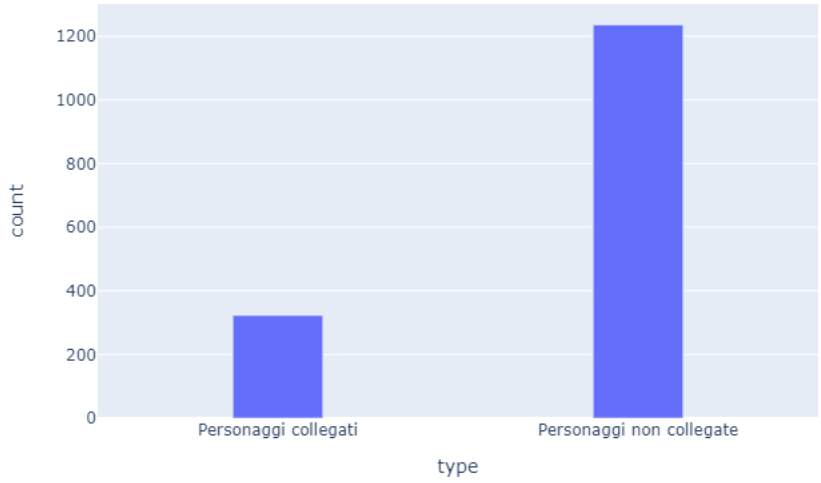

Figure 2: Grafico al numero di personaggi che sono collegabili con i dati ottenuti tramite web scraping.

3.2.2 Descrizione fumetti

Le analisi di completezza effettuate sulle descrizioni dei fumetti, ottenute tramite web api, consiste nel vedere quanti fumetti possiedono una descrizione. In particolare circa 19000 fumetti non hanno descrizione al netto dei circa 50000 fumetti.

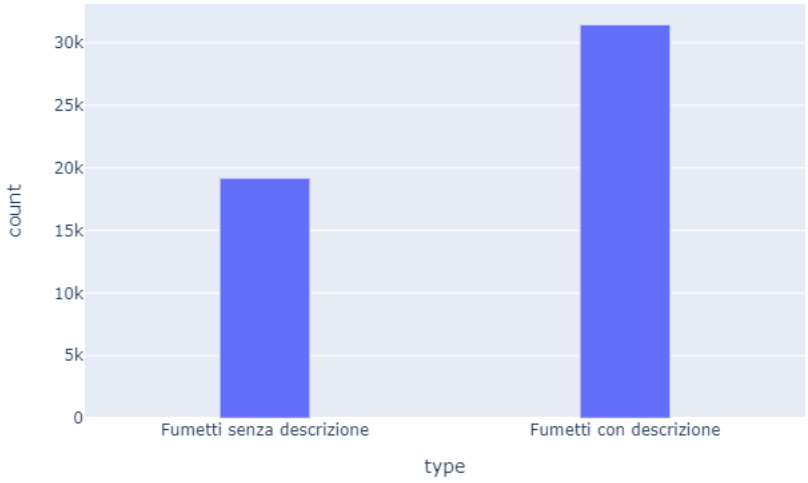

Figure 3: Grafico relativo al numero di fumetti con e senza descrizione

4 DATA INTEGRATION

Per il processo di Data Integration sono state collegate dove possibile le due sorgeti dati tramite gli schemi dei personaggi, i quali fungono da punto di incontro fra le due sorgenti dati. è stata realizzata un operazione di record linkage fra i dati dei personaggi

4.1 Data preparation

Prima di tutto si aggiunge una nuova colonna al dataset del web scraping e della web api, contenente il nome del personaggio senza alcuni caratteri particolari come ad esempio " o ’ e senza qualsiasi porzione di testo racchiusa fra delle parentesi. Queste operazioni sono state effettuate tramite l’utilizzo della libreria python "re" per pulire le stringhe. Le parentesi sono tolte per poter fare il confronto dei personaggi senza tenere conto in questo caso della singola variante del personaggio(ad esempio stesso personaggio di universi differenti). In output si avranno i due schemi relativi ai personaggi con un campo nome processato come detto precedentemente.

4.2 Schema transformation

Durante la fase di schema matching è stato effettuato un processo di reverse Engineering per capire cosa rappresentassero i dati della web api, non è stato fatto per lo scraping in quanto abbiamo scelto noi

quali dati prendere e quindi lo schema era già chiaro a priori L'analisi dello schema relativo ai personaggi è così composto:

1. Id: identificativo univoco assegnato dall'api ad ogni personaggio
2. description: Breve descrizione del personaggio
3. modified: ultima modifica apportata al personaggio
4. thumbnail: url che porta ad un immagine del personaggio
5. comics: lista di fumetti in cui il personaggio è presente, con relativo id fumetto
6. series: lista di serie di fumetti in cui il personaggio è presente
7. stories: lista di storie in cui il personaggio è presente
8. events: lista di eventi in cui ha partecipato il personaggio

L'analisi dello schema relativo ai fumetti ottenuti tramite web api ha permesso di identificare i seguenti attributi:

1. Id: identificativo univoco relativo al fumetto
2. title: titolo del fumetto
3. description: breve descrizione del fumetto
4. modified: ultima modifica dei dati relativi al fumetto
5. isbn, upc, diamondCode, ean, issn: identificativi editoriali relativi al fumetto
6. forma: Formato di stampa del fumetto
7. pageCount: numero di pagine del fumetto
8. series: lista delle serie a cui appartiene il fumetto
9. thumbnail: url relativa ad un immagine della copertina del fumetto
10. images: url relativi a varie immagini del fumetto
11. creators: lista dei creatori del fumetto
12. characters: lista di personaggi appartenenti al fumetto
13. stories: lista di storie appartenenti al fumetto
14. events: lista di eventi significativi presenti nel fumetto

4.3 Corrispondences Investigation

Durante la fase di *Corrispondences Investigation*, si è notato che si possono unire le due basi di dati sulla base dei personaggi, più specificatamente è possibile stabilire un collegamento fra i personaggi sulla base del loro nome. Dopo varie analisi si è notato che sia nel caso della base di dati ottenuta tramite consultazione della web api sia di quella ottenuta tramite web scraping, che un personaggio è identificato tramite un nome privo di parentesi e un'eventuale variante del personaggio è identificata da un nome all'interno della parentesi. Alla luce di queste considerazioni una regola di confronto fra i due schemi consiste nel vedere se due nomi di personaggi sono uguali dopo aver rimosso parentesi ed eventuali testi all'interno delle parentesi.

4.4 Schemas Integration

La fase di schemas integration avviene durante l'inserimento dei dati nel database: per ogni record presenti nei due schemi relativi ai personaggi vengono rimosse le porzioni di stringa contenute all'interno di parentesi tramite l'utilizzo della libreria *re*; in seguito viene creato all'interno del database un nodo di tipo *character* se non è già presente un nodo dello stesso tipo con lo stesso nome. Al nodo character, sia che è stato creato oppure no, viene collegato un nodo *Character Variant* identificato tramite nome completo, ovvero il nome prima della rimozione delle parentesi, contenente tutte le informazioni presenti nel record personaggio in esame. Ovviamente anche il nodo *Character Variant* viene creato solo se non esiste già, mentre se esiste vengono solo aggiunte informazioni mancanti presenti nella base di dati da cui proviene il personaggio in esame.

5 DATABASE

Il database scelto per lo starage dei dati è un database a grafo. i motivi che hanno portato a questa scelta sono:

1. Dati non in real time: la mole di dati riguardante l'universo marvel non è enorme e non necessita di un inserimento di dati real time, quindi le operazioni di scrittura sono relativamente poche e le operazioni in lettura sono molte di più; proprio per questo un database a grafo può essere utilizzato in quanto lento in scrittura ma veloce in lettura.

2. Lo scopo del dataset è mostrare le varie relazioni fra i prodotti marvel e quindi un database a grafo è l'ideale, in quanto è basato sul concetto di relazioni e collegamenti

5.1 Struttura Database

La struttura del database è la seguente:

5.1.1 Nodi

I nodi presenti all'interno del database sono:

1. Comic: nodo rappresentante un fumetto Marvel, con al suo interno il suo id fumetto ottenuto dall'api, i vari codici di identificazione,come ad esempio ISBN, la descrizione del fumetto se è presente, il formato del fumetto e il numero di pagine.
2. Movie: nodo rappresentante un film Marvel, con al suo interno i vari dati relativi al film come ad esempio la trama del film, i registi, scrittori, produttori, compositori, incassi, durata del film, e data di rilascio del film
3. Tv Show: nodo rappresentante una serie tv Marvel, con al suo interno i vari dati relativi al film come ad esempio la trama della prima stagione, i registi, gli showrunner, produttori, compositori.
4. Character: nodo rappresentante la versione generica di un personaggio, contiene al suo interno il nome del personaggio
5. Character Variant: Nodo che rappresenta una variante precisa di un personaggio, contenente informazioni come ad esempio il nome, una descrizione del personaggio e una biografia del personaggio.

5.1.2 Relazioni

Le relazioni presenti fra i nodi all'interno del database sono:

1. In Film: relazione che collega un nodo «Character Variant» ad un nodo film, rappresenta appunto la partecipazione del personaggio all'interno del film
2. In Serie: relazione che collega un nodo «Character Variant» ad un nodo ts show, rappresenta la partecipazione del personaggio all'interno di una serie tv.
3. In fumetto: relazione che collega un nodo «Character Variant» ad un nodo «Comic», rappresenta la partecipazione del personaggio all'interno di un fumetto.
4. Conosce: relazione che collega un nodo «Character Variant» ad un nodo «Character Variant», rappresenta una relazione interpersonale fra due personaggi.

5. «Variante di»: relazione che rappresenta una variante di un personaggio base, collega un nodo di tipo «character» ad un nodo di tipo «character variant».

5.2 Statistiche database

All'interno del database sono stati creati in totale 60411 nodi distribuiti nel seguente modo:

Etichetta Nodo	Numero Di Nodi
Character	4609
Character Variant	5113
Comic	50607
Movie	43
Tv Show	19

Fra i nodi del database sono state create 117763 relazioni distribuite nel modo seguente

Etichetta Relazione	Numero Di Relazioni
Conosce	19040
In Film	531
In Serie	2405
In Fumetto	90503
Variante di	5284

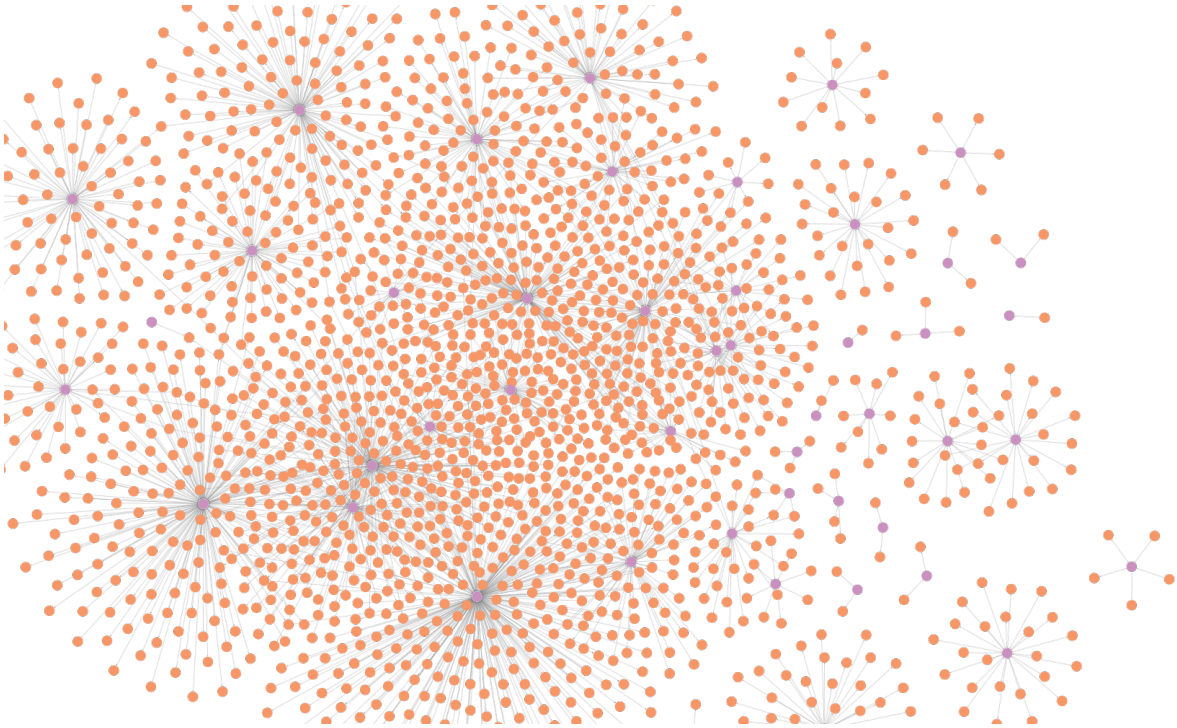

Figure 4: Visualizzazione di alcuni nodi Character variant(Viola) in relazione a dei nodi Comic(Arancione)

5.3 Alcune Query

Grazie alla struttura a grafo del database è possibile ottenere facilmente tramite query relazioni fra i nodi. Ad esempio potremmo trovare i personaggi conosciuti dal personaggio "Iron Man", che sono presenti nel film *Avengers*. Per ottenere questo risultato si può utilizzare la seguente query CYPHER:

MATCH (i:character_variant)-[r:conosce]->(c:character_variant)-[p:in_film] ->(m:movie) WHERE i.name = "Iron Man" AND m.title = "The Avengers" RETURN i,r,c,p,m

REFERENCES