Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

Rome’s treasures unveiled:  
an insider’s journey

Gruppo di lavoro

* Daniele Guerra, 744399, d.guerra6@studenti.uniba.it
* Nicola Lassandro, 735968, n.lassandro4@studenti.uniba.it

Repository GitHub

<https://github.com/NicolaLassandro/progetto_icon_guerra_lassandro>

AA 2022-2023

Introduzione

Il software lavora nel dominio degli itinerari turistici, in particolare (al momento) è dedicato alla città di Roma.

Esso mette a disposizione dell’utente una panoramica completa di ciò che la città ha da offrire, garantendo una flessibilità che permette di personalizzare la propria esperienza in base alle proprie preferenze ed esigenze.

L’obiettivo del software è quello di individuare il percorso di visita migliore, principalmente sulla base della distanza percorsa, si tiene anche conto dei limiti di tempo e budget impostati, tutto ciò garantendo una qualità di punti di interesse visitati elevata.

Sommario

[Gruppo di lavoro 1](#_Toc137745240)

[Repository GitHub 1](#_Toc137745241)

[Introduzione 2](#_Toc137745242)

[Strutturazione del progetto 4](#_Toc137745243)

[Elenco argomenti di interesse 5](#_Toc137745244)

[Rappresentazione della conoscenza mediante Kb 6](#_Toc137745245)

[Sommario 6](#_Toc137745246)

[Strumenti utilizzati 7](#_Toc137745247)

[Decisioni di progetto 8](#_Toc137745248)

[Valutazioni 8](#_Toc137745249)

[Problema di ricerca su grafo 9](#_Toc137745250)

[Sommario 9](#_Toc137745251)

[Strumenti utilizzati 9](#_Toc137745252)

[Decisioni di progetto 9](#_Toc137745253)

[Valutazioni 9](#_Toc137745254)

[Apprendimento supervisionato (regressione) 10](#_Toc137745255)

[Sommario 10](#_Toc137745256)

[Strumenti utilizzati 10](#_Toc137745257)

[Decisioni di progetto 10](#_Toc137745258)

[Valutazioni 10](#_Toc137745259)

[Conclusioni e sviluppi futuri 11](#_Toc137745260)

[Riferimenti bibliografici 12](#_Toc137745261)

Strutturazione del progetto

* Cartella Knowledge: contiene tutti i file prolog e python relativi alla rappresentazione della conoscenza nella Kb utilizzata nel progetto.
  + Facts.pl
  + KbManager.py
  + Landmark.py
  + Preprocessor.py
  + Rules.pl
  + RuntimeFacts.pl
  + Utility.py
* Cartella Learning: contiene tutti i file python relativi alla fase di apprendimento supervisionato.
  + FeedbackGenerator.py
  + Kfold.py
  + Knn.py
  + ModelInitializer.py
  + PreProcessorLearning.py
  + RegressionTree.py
* Cartella Logs: contiene tutti i file testuali di log utilizzati per supervisionare l’andamento degli script di tutte le altre cartelle.
* Cartella Search: contiene tutti i file python utilizzati nel problema di ricerca su grafo.
  + Cartella Libs: contiene gli algoritmi predefiniti che sono stati utilizzati come base per il problema di ricerca realizzato dal gruppo.
  + ItinerarySearchProblem.py
  + **MainSearch.py**: .
  + NodeGraph.py
* Cartella Storage: contiene tutti i file serializzati in formato pickle.

Elenco argomenti di interesse

Di seguito sono presentati i macro-argomenti trattati durante lo sviluppo dell’applicativo.

* Rappresentazione della conoscenza mediante Kb
* Problema di ricerca su grafo
* Apprendimento supervisionato (regressione)

In fase di scelta dell’idea progettuale e della sua realizzazione si è cercato di includere la più vasta gamma di argomenti trattati durante il corso di Ingegneria della conoscenza.

Rappresentazione della conoscenza mediante Kb

Sommario

La prima operazione svolta durante lo sviluppo dell’applicativo turistico è stata la raccolta dei dati relativi ai punti di interesse della città di Roma, i quali sono stati utilizzati per l’inizializzazione delle istanze della classe di riferimento: Landmark.  
Gli attributi presenti in tale classe sono:

* placeId, name, address, type, properties, lat, lon, age.  
  Questi rappresentano le caratteristiche “anagrafiche” del luogo di interesse.
* rating, ratingCount.  
  Questi illustrano la popolarità e la qualità del punto di interesse sulla base dei giudizi degli utenti.
* centreDistance.  
  E’ la distanza da quello che è stato definito arbitrariamente il punto centrale del turismo della città.
* tourismRate, price.  
  Questi fanno riferimento alle carattistiche “turistiche”, le quali sono fattori influenzanti della visitabilità del luogo.
* handicapAccessibility, surface, height.  
  Questi sono gli attributi strutturali del punto di interesse.

Tutti gli altri campi presenti verranno popolati solo in un secondo momento.

Successivamente queste istanze sono state memorizzate all’interno di un dizionario e tutte le feature sono state immagazzinate all’interno di una base di conoscenza sotto forma di fatti. La maggior parte di essi seguivano uno stesso formato: feature(landmark\_name, feature\_value).

Di seguito sono presentati alcuni esempi.  
Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

La feature distance fa eccezione dal formato Prolog delle altre, essa è così strutturata: feature(landmark\_1\_name, landmark\_2\_name, feature\_value).  
Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente  
Questa feature è di fondamentale importanza all’interno del progetto in quanto si trova alla base del modulo di ricerca su grafo, nucleo del caso di studio.  
Per questo motivo è stato necessario conservare tutti i fatti ad essa relativi nell’apposito file Prolog, cosa che non è avvenuta per tutte le feature.

La KB, dopo essere stata arricchita con diverse regole, è stata interrogata per generare nuova conoscenza.

Tra le regole realizzate ce ne sono alcune particolarmente importanti:

* calculateDensity, che permette di definire la densità turistica del luogo, intesa come media delle distanze da tutti gli altri posti registrati (prelevati precedentemente mediante il predicato findall).  
  Immagine che contiene testo, schermata, software, computer

  Descrizione generata automaticamente
* calculateTourismPriority, serve a calcolare la priorità di visità del punto di interesse sulla base delle altre feature che lo caratterizzano.  
  Ognuna di esse influenza in modo diverso il risultato finale, che viene successivamente salvato nel dizionario e utilizzato nel modulo di ricerca su grafo per garantire una soglia minima di qualità dei luoghi previsti dall’itinerario turistico.  
  Immagine che contiene testo, schermata, software, computer

  Descrizione generata automaticamente
* calculateTimeToVisit, si occupa di stabilire il tempo necessario alla visita di un luogo, questo è influenzato sia da aspetti strutturali (come le dimensioni) sia da aspetti turistici (all’aumentare del tasso di turismo, il luogo risulta più affollato).  
  Immagine che contiene testo, schermata, software, computer

  Descrizione generata automaticamente

Di seguito è presentato un pezzo di codice python con il quale viene interrogata la base di conoscenza per ciascuno dei punti di interesse registrati, i risultati ricevuti vengono posti in un contenitore (result) e successivemente estratti e assegnati all’omonimo attributo dell’istanza di Landmark corrispondente nel dizionario.

Immagine che contiene elettronica, testo, computer, schermata

Descrizione generata automaticamente

Anche le interrogazioni che restituiscono risultati di diverso tipo (ad esempio booleani) seguono un ragionamento analogo.

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

Infine, il dizionario contenente tutte le istanze di Landmark è stato aggiornato, con tutti i valori delle nuove feature a disposizione, e serializzato per poter essere utilizzato nei successivi moduli del progetto.

Strumenti utilizzati

La raccolta dei dati utili è stata effettuata mediante l’impiego di una API a pagamento messa a disposizione da Google Places (la piattaforma di Google Maps): **nearbysearch**, essa consente di recuperare una lista di luoghi nelle vicinanze di una determinata posizione geografica.

Per il suo utilizzo è stato necessario generare (gratuitamente) una API-Key della durata di pochi mesi ed è stata impiegata la **libreria Requests** fornita da Python.

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente  
La chiamata richiede diversi parametri per restituire i risultati corretti, di seguito sono citati quelli impiegati nel corrente caso di studio:

* API-Key
* Le coordinate di latitudine e longitudine del centro di ricerca, a partire dal quale sono stati individuati i risultati.
* Il raggio massimo di estensione della ricerca dei luoghi.
* Il tipo dei luoghi, espresso mediante una parola chiave, sulla base del quale viene effettuto il filtraggio, allo scopo di ottenere risultati specifici.

La risposta risultante della chiamata viene restituita in formato json e contiene al suo interno numerosi attributi, tra questi ne sono stati selezionati ed estratti alcuni utili allo scopo dell’applicativo:

* place\_id, un identificativo che viene assegnato ad ogni elemento registrato sulla piattaforma Google.
* user\_ratings\_total, un intero che rappresenta la quantità di recensioni degli utenti relative all’elemento.
* name, il nome del luogo.
* geometry.location.lat e geometry.location.lon, le coordinate spaziali del luogo in formato decimale.
* rating, la media delle votazioni (da 1 a 5) contenute nelle recensioni.

Le restanti caratteristiche dei punti di interesse, come suggerito dal , per via della difficoltà di reperimento, sono state generate casualmente mediante la **libreria random** messa a disposione da python. Ovviamente la generazione casuale ha tenuto conto di intervalli verosimili dei valori delle feature.

Per la rappresentazione della conoscenza e la sua successiva interrogazione si è deciso di utilizzare il **linguaggio Prolog**, un linguaggio di programmazione logico basato su predicati, che rappresentano fatti o relazioni tra oggetti. Un programma Prolog è costituito da un insieme di predicati e regole, queste ultime vengono utilizzate per derivare nuovi fatti o per risolvere interrogazioni sulla base di conoscenza.

Decisioni di progetto

spiegare le chiamate multiple

spiegare in base a cosa è stato scelto il “centro turistico esatto di roma”

spiegare perché abbiamo usato write e read invece di assertz

Problema di ricerca su grafo

Sommario

Strumenti utilizzati

Decisioni di progetto

Valutazioni

Apprendimento supervisionato (regressione)

Sommario

Strumenti utilizzati

Decisioni di progetto

Valutazioni

Conclusioni e sviluppi futuri

Riferimenti bibliografici

https://developers.google.com/maps/documentation/places/web-service/search-nearby?hl=it