Progetto prerequisiti

Andrea Mentasti

Estrazione relazione di prerequisito

L'obiettivo del progetto è quello di definire una metodologia per l'individuazione della **relazione di prerequisito tra due Learning Objects (LOs).**

Una relazione di prerequisito è una relazione pedagogica che indica l'ordine in cui i concetti possono essere proposti all'utente. Formalmente diciamo che esiste una relazione di prerequisito tra due concetti se uno è significativamente utile per comprendere l'altro.

Si tratta di un problema di classificazione binaria, dove 1 indica che B è prerequisito di A, e 0 indica la non presenza della relazione.

Qui viene mostrato un esempio del funzionamento: nella prima riga il concetto B "Luce" è prerequisito di A "Riflessione interna", mentre nella seconda riga non c'è questa relazione.

```
Riflessione interna totale,Luce,1
Plasticità (fisica),Durezza,0
Fisica,Accelerazione di gravità,0
...
Campo magnetico,Magnete,1
```

Dataset e preprocessing

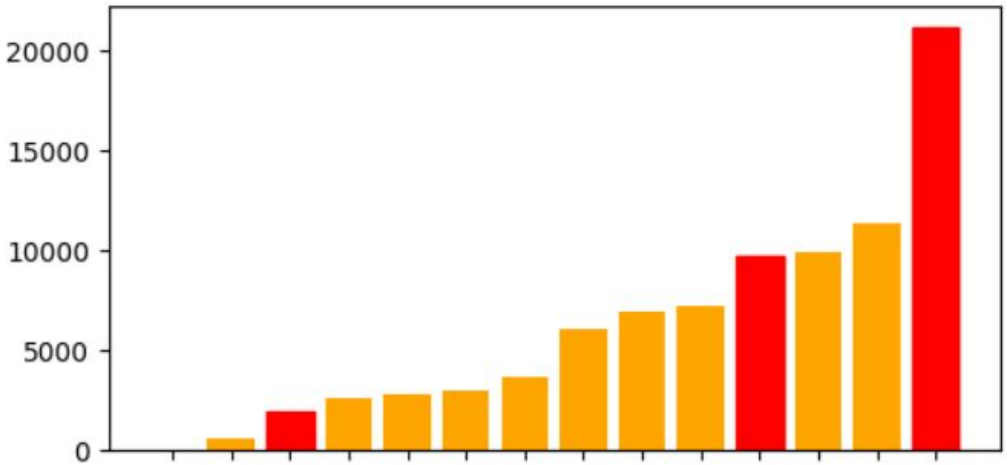
DATASET MERLOT (Multimedia Education Resource for Learning & Online Teaching)
e **DATASET OER** (Open Educational Resources)

1) Preprocessing per unire i due dataset
2) Filtraggio per language = English
3) **Filtraggio per disciplina informatica** per valutare i topic più specifici

Dataset con risorse didattiche con:

- URI
- **domain level**
- title
- **description**
- type
- upload date
- **language**
- difficulty
- format
- duration
- destination_public
- min/max age

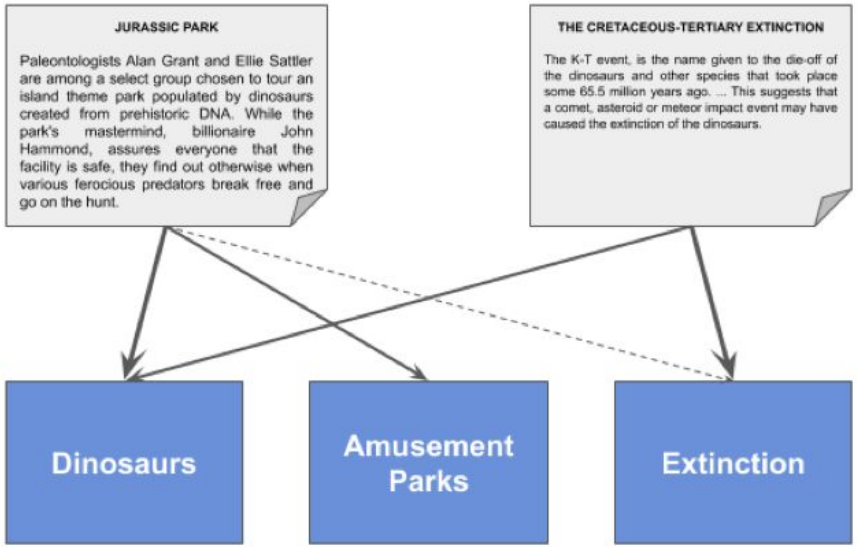
Dataset e preprocessing



Le risorse in lingua inglese sono 86947, ossia **88.18%** di tutte le risorse.

Nel grafico vengono mostrate le discipline tenute in considerazione nel nostro dataset. In ordine crescente: informatica 21,140, matematica 9,766 e ingegneria 1,947.
Abbiamo quindi usato 32,853 risorse, cioè il **37,78%** del dataset in lingua inglese.

Estrazione Topics



Il primo step che è stato svolto riguarda l'estrazione dei topic rappresentativi dei Learning Objects. Per fare questo è stato utilizzato un approccio di **Topic Modeling**, un metodo di machine learning non supervisionato che riceve in input un corpus di documenti e restituisce gli argomenti e i concetti principali.

Esistono molteplici algoritmi di topic modeling, ma la scelta in questo caso è ricaduta su **Latent Dirichlet Allocation** date le ottime performance in termini di coerenza semantica degli argomenti estratti.

Latent Dirichlet Allocation (LDA)

La **Latent Dirichlet Allocation (LDA)** è uno degli approcci di topic modeling utilizzato nel Natural Language Processing (NLP) per scoprire gli argomenti sottostanti che sono presenti in una raccolta di documenti di testo.

Presupposti

- ogni documento è una miscela di un numero fisso di topic
- ogni topic è una distribuzione di probabilità sulle parole

1 Inizializza

Assegnazione casuale dei topic ad ogni parola di ogni documento

2 Aggiorna

Aggiornamento del topic assegnato alla singola parola in base ai topic nel documento e alle parole nei topic

3 Ripeti

Ripetizione del secondo step su tutte le parole

4 Itera

Iterando, le parole andranno verso topic “migliori”

Implementazione LDA

Abbiamo applicato **Latent Dirichlet Allocation ad ogni descrizione** per estrarre **5 topics composti di 5 parole**. Di questi 5 topics abbiamo mantenuto il primo, composto dalle 5 parole significative.

La fase successiva prevede l'estrazione delle pagine di wikipedia e poi la conseguente valutazione della similarità.

	title	text_preprocessed	concept1	concept2	concept3	concept4	concept5
0	Holiday STEM Activities for Kids	this resource includes activities that can int...	day	christmas	activities	thanksgiving	resource
1	Blockly Games	blockly games is a series of educational games...	games	computer	designed	collaborate	blockly
2	Day of AI	this resource is to support teachers and educa...	day	engage	resource	participants	ai
3	Student Investigation Guide for Introduction t...	in this book, you are going to engage in a ser...	investigations	question	group	investigation	small
4	Bridges to STEM Learning for Grades K-8	this course provides knowledge and skills in s...	engaging	throughout	stem	engineering	math

Estrazione pagine da wikipedia

Per l'estrazione delle pagine wikipedia abbiamo usato la libreria wikipedia-api di python, la quale ci permette di estrarre le descrizioni delle parole da Wikipedia.

L'output consiste nel dataset precedente con 5 colonne aggiuntive che rappresentano le 5 descrizioni da wikipedia.

```
import wikipediaapi

wiki_wiki = wikipediaapi.Wikipedia('en')

wiki_concept1 = []

for word in concept1:
  try:
    page = wiki_wiki.page(word)
    page_sum = page.summary
    wiki_concept1.append(page_sum)
  except:
    wiki = None
    wiki_concept1.append(wiki)
```

Valutazione similarità LOs e descrizioni dei topics

Fino a questo momento le operazioni svolte sul dataset sono state l'applicazione dei LDA per fare topic modeling e l'estrazione da wikipedia delle descrizioni dei concetti associati a ogni singolo LO.

La fase successiva prevede l'identificazione della pagina Wikipedia più simile alla descrizione del Learning Object. Per fare ciò si poteva procedere in due direzioni:

- valutare la similarità del LO rispetto alle 5 wikipedia ad esso associate.
- valutare la similarità del LO rispetto a tutte le descrizioni di Wikipedia estratte.

Date le dimensioni, sarebbe stato troppo time consuming effettuare il confronto tra ogni risorsa e le 5 descrizioni a essa assegnate usando sentence transformer.
Per questo motivo abbiamo implementato una ricerca tramite clustering Approximate Nearest Neighbours.

Approximate Nearest Neighbour (ANN)

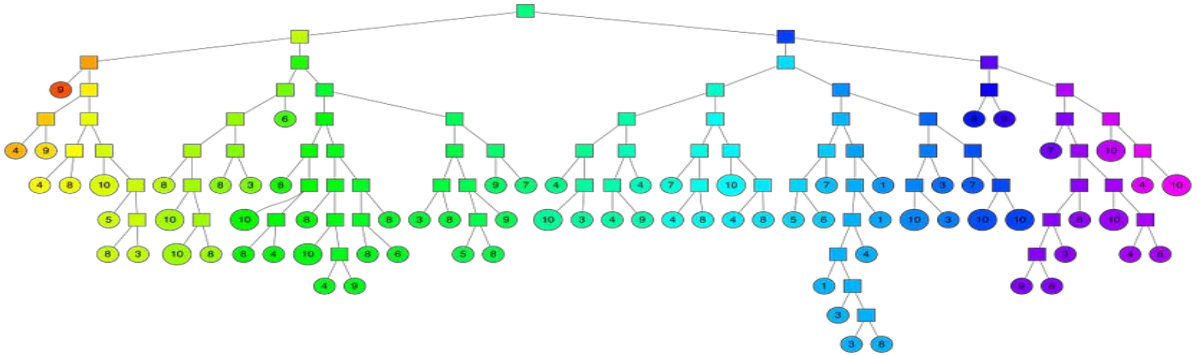
Approximate Nearest Neighbour (ANN) è un metodo di ricerca dei vettori più simili, e viene preferito rispetto a un normale Exact Nearest Neighbour in casi di dataset di grandi dimensioni. I passaggi di ANN sono i seguenti:

- **Trasformazione vettoriale**: tra cui la riduzione della dimensionalità e la rotazione del vettore.
- **Vector Encoding**: applicata ai vettori per costruire l'indice effettivo per la ricerca. Per questa fase abbiamo usato un metodo di **encoding basato su Trees**. Ogni albero è costruito nel modo seguente, scegliamo due punti a caso e dividiamo lo spazio in due, continuiamo a suddividere i sottospazi in modo ricorsivo finché i punti associati a un nodo non sono abbastanza piccoli. La foresta viene attraversata per ottenere un insieme di punti che sia il più vicino possibile al punto che ci interessa.
- **None Exhaustive Search Component**: applicato ai vettori per evitare la ricerca esaustiva.

Approximate Nearest Neighbour (ANN)

ANN funziona grazie alla suddivisione dello spazio in strutture ad albero. Ogni ramo viene generato partendo da una divisione dello spazio basata su due elementi casuali. Ogni volta che un nuovo vettore viene introdotto, questo percorrerà i rami dell'albero fino ad arrivare su un ramo contenente le altre osservazioni ad esso più simili.

Gli iperparametri da scegliere sono il numero di alberi e il numero di elementi nei rami terminali. Più alberi garantiscono una maggiore recall ma maggiori tempi computazionali.



Applicazione Approximate Nearest Neighbours

La libreria scelta per eseguire il clustering tramite ANN è Annoy, una libreria creata specificatamente per questa operazione.

Dopo la prima fase di encoding, ANN procede con la ricerca dei punti più vicini tramite il comando **annoy_index.get_nns_by_vector()**

Successivamente i risultati più pertinenti e il loro score di similarità vengono inseriti nel dataset.

```
print("Corpus loaded with {} sentences / embeddings".format(len(corpus_sentences

for index, row in dataset.iterrows():
    #dataset.at[index, 'embedding'] = embedding

    inp_question = row['text_preprocessed']
    question_embedding = model.encode(inp_question)

    corpus_ids, scores = annoy_index.get_nns_by_vector(question_embedding,
                                                       top_k_hits,
                                                       include_distances=True)
    hits = []

    for id, score in zip(corpus_ids, scores):
        hits.append({'corpus_id': id, 'score': 1-((score**2) / 2)})

    for hit in hits[0:top_k_hits]:
        dataset.at[index, 'best_ANN'] = corpus_sentences[hit['corpus_id']]
        dataset.at[index, 'best_ANN_cos_sim'] = hit['score']
```

Parametri di Approximate Nearest Neighbour (ANN)

La configurazione di iperparametri migliore in questo caso è quella di utilizzare **1024 alberi**. In questo caso la media delle similarità calcolate tra i LOs e le wikipedia associate è di **0.476**. In questo caso il numero di elementi in ogni ramo terminale è 1, quindi ogni LO viene confrontato con il suo vicino più simile.

La misura di similarità scelta è la **Cosine Similarity**, una metrica utilizzata per misurare la somiglianza dei documenti indipendentemente dalle loro dimensioni. Misura il coseno dell'angolo tra due vettori proiettati in uno spazio multidimensionale.

	stat	dataset-128 trees	dataset-256 trees	dataset-512 trees	dataset-1024 trees
0	Media	0.440	0.451	0.467	0.476
1	Mediana	0.431	0.443	0.460	0.468
2	Dimensione DF	5084.000	4956.000	4834.000	4771.000

Dataset e valutazione risultati

	title	text_preprocessed	best_ANN	best_ANN_cos_sim
0	Holiday STEM Activities for Kids	this resource includes activities that can int...	Spongelab is a science education website for t...	0.462783
1	Blockly Games	blockly games is a series of educational games...	Blockly is a client-side library for the progr...	0.645902
2	Day of AI	this resource is to support teachers and educa...	Creativity is a phenomenon whereby something n...	0.419912
3	Student Investigation Guide for Introduction t...	in this book, you are going to engage in a ser...	A planet is a large, rounded astronomical body...	0.567468
4	Bridges to STEM Learning for Grades K-8	this course provides knowledge and skills in s...	OpenStax (formerly OpenStax College) is a nonp...	0.400543

	stat	Dataset 0.40-0.45	Dataset 0.45-0.50	Dataset 0.50-0.55	Dataset 0.55-0.60	Dataset 0.6 -
0	Media	0.426	0.474	0.524	0.573	0.657
1	Mediana	0.426	0.473	0.523	0.572	0.642
2	Max	0.450	0.500	0.550	0.600	0.931
3	Standard Deviation	0.014	0.014	0.014	0.014	0.053
4	Dimensione DF	6415.000	6496.000	4871.000	2883.000	2626.000

Dataset e valutazione risultati

I risultati della Cosine Similarity mostrano che per la parte di dataset composta da LOs e descrizioni di wikipedia con similarità superiore a 0.6 sono presenti 2626 risultati.

Per la fase successiva si possono quindi utilizzare queste risorse, in modo da valutare il modello su quei LOs su cui è stato possibile trovare una descrizione di wikipedia più simile possibile.

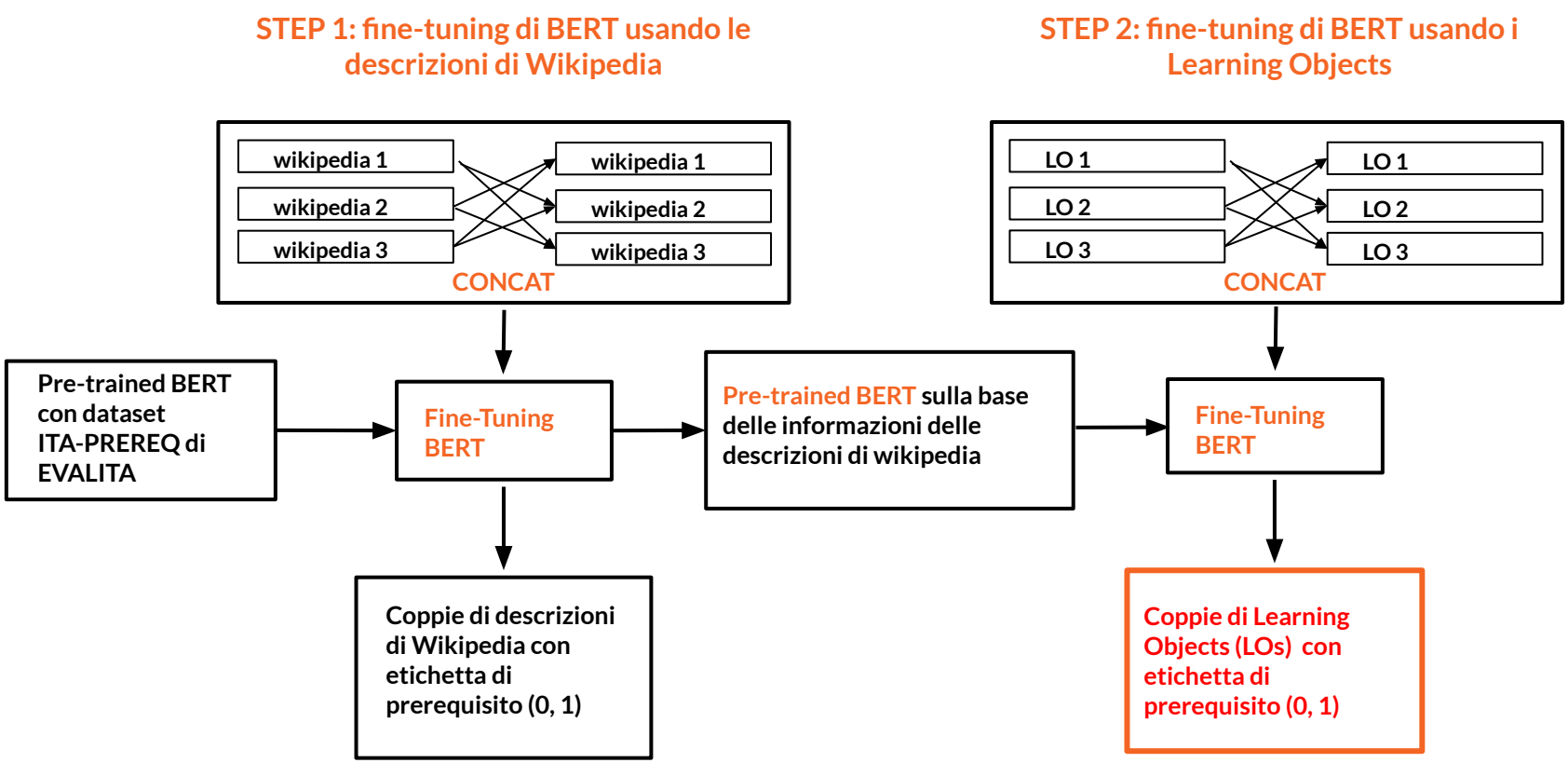
	stat	Dataset 0.40-0.45	Dataset 0.45-0.50	Dataset 0.50-0.55	Dataset 0.55-0.60	Dataset 0.6 -
0	Media	0.426	0.474	0.524	0.573	0.657
1	Mediana	0.426	0.473	0.523	0.572	0.642
2	Max	0.450	0.500	0.550	0.600	0.931
3	Standard Deviation	0.014	0.014	0.014	0.014	0.053
4	Dimensione DF	6415.000	6496.000	4871.000	2883.000	2626.000

Struttura dell'approccio proposto

Per la fase successiva l'approccio proposto è simile a quello effettuato dal team di Angel et AL. nella task di individuazione dei prerequisiti a EVALITA 2020. Come in questo caso, l'obiettivo del team è stato quello di trovare una metodologia per l'individuazione della relazione di prerequisito tra delle coppie di concetti.

Nelle successive fasi il nostro approccio è quello di replicare la metodologia proposta da Angel et al., la quale consiste in una **Neural Network Single Layer** addestrata usando delle rappresentazioni di coppie di concetti estratte usando un **BERT italian model** sul quale è stato eseguito un **fine-tuning** usando il dataset ITA-PREREQ, il quale contiene coppie di concetti italiani legati da una relazione di prerequisiti.

Allo stesso modo, nel nostro caso prevediamo di effettuare un fine-tuning di BERT usando le coppie di testi di wikipedia estratti e ricevere in output label 0 e 1. Successivamente utilizzare questo modello pre-addestrato ed estenderlo alle coppie di learning objects per prevedere la relazione di prerequisito.


STEP 1: fine-tuning di BERT usando le descrizioni di Wikipedia
STEP 2: fine-tuning di BERT usando i Learning Objects
wikipedia 1
wikipedia 2
wikipedia 3
wikipedia 1
wikipedia 2
wikipedia 3
CONCAT
LO 1
LO 2
LO 3
LO 1
LO 2
LO 3
CONCAT
Pre-trained BERT con dataset ITA-PREREQ di EVALITA
Fine-Tuning BERT
Pre-trained BERT sulla base delle informazioni delle descrizioni di wikipedia
Fine-Tuning BERT
Coppie di descrizioni di Wikipedia con etichetta di prerequisito (0, 1)
Coppie di Learning Objects (LOs) con etichetta di prerequisito (0, 1)