POLITECNICO DI TORINO

Corso di Laurea Magistrale in INGEGNERIA DEL CINEMA E DEI MEZZI DI COMUNICAZIONE

Politecnico di Torino

Video-to-text e text-to-video retrieval: un approccio basato sulle knowledge base

Relatrice:

Prof. Laura FARINETTI

Correlatore:

Dott. Lorenzo CANALE

Candidata

Elisa PLACENTINO

Aprile 2022

Sommario

Il grande aumento dei documenti a disposizione ha reso necessario l'introduzione di sistemi per il veloce e preciso reperimento degli stessi. Oggi la disponibilità di contenuti multimediali e il progresso dei sistemi informatici hanno permesso di esplorare il campo del cross-media retrieval, ovvero il recupero di materiale a partire da un documento di query di diversa natura. La difficoltà nel reperire materiali eterogenei deriva dalla diversa modalità di rappresentazione dei contenuti e dall'impossibilità di associare i documenti in modo diretto.

Questa tesi esplora i task di video-to-text retrieval e text-to-video retrieval ovvero la ricerca di documenti testuali a partire da una query rappresentata da un video e viceversa. La maggior parte dei metodi attualmente in uso si fondano sull'utilizzo di reti neurali che presentano architetture complesse e richiedono numerose risorse computazionali. In questo elaborato si indaga una nuova metodologia che si prepone di rappresentare sia i video sia i testi come insiemi di named entity linkate con la knowledge base Wikidata. In particolare, viene posta maggiore attenzione sull'estrazione delle entità dai video, testando vari sistemi di object detection per il riconoscimento degli oggetti presenti nei frame e un mapping manuale tra le label estratte e le entità di Wikidata.

I risultati dei ranking ottenuti con i sistemi di object detection vengono confrontati con una baseline costituita da un ordinamento random e un competitor, la Google Vision API.

I valori ricavati permettono di evidenziare che il task beneficia dell'approccio intrapreso soprattutto per ciò che concerne il recupero di contenuti video a partire da query testuali.

Ringraziamenti

Questa tesi è la conclusione di un percorso che mi ha permesso di imparare molto, non solo a livello nozionistico, ma anche a livello umano. Ringrazio quindi i professori e i colleghi con cui ho collaborato nei vari progetti di gruppo proposti in questi anni.

Grazie alla mia famiglia per avermi supportata e per non avermi fatto mancare nulla.

Un pensiero speciale va ai miei amici incontrati all'università che hanno condiviso con me intere giornate o piccoli momenti e a tutti coloro che in questi anni hanno contribuito con la loro presenza a rendermi quella che sono oggi.

Tabella dei Contenuti

Elenco delle tabelle

Elenco delle figure

Acronimi

VTR

Video-Text Retrieval

IR

Information Retrieval

DNN

Deep Neural Network

RNN

Recurrent Neural Network

POOL

Pooling Layer

CONV

Convolution Layer

FC

Fully Connected Layer

ReLU

Rectified Linear Unit

FFNN

Feed Forward Neural Network

NLP

Natural Language Processing

BERT

Bidirectional Encoder Rapresentations from Transformers

ViT

Vision Transformer

CLIP

Contrastive Language-Image Pre-training

Capitolo 1

Introduzione

Il veloce e preciso reperimento di documenti presenti in un database tali da rispondere in maniera puntuale alla richiesta effettuata dall'utente interessa da molto tempo il campo della ricerca.

Coerentemente a quanto riportato in [1], da un punto di visto storico l'idea dell'accessibilità delle informazioni contenute nei testi delle biblioteche del mondo, si divulga a partire dal 1945, quando Vannevar Bush in un articolo [2] invita gli scienziati a reindirizzare il proprio lavoro intrapreso nel periodo di guerra per scopi pacifici. Egli sostiene che la condivisione del materiale di ricerca gioverebbe al mondo scientifico, ma una grande quantità di contenuti condivisi su scala globale necessita di un sistema che permetta il rapido e puntuale accesso.
Nel periodo in cui viene scritto l'articolo gli strumenti tecnologici non permettono la realizzazione di questa idea, ma nei decenni successivi i miglioramenti della strumentazione e la popolarizzazione del campo di studio ne consentono considerevoli miglioramenti. La crescita di interesse nel campo dell'information retrieval comincia nel 1960, quando inziano a tenersi numerose conferenze su tale argomento. In questo decennio il crescente numero di termini da indicizzare e recuperare manualmente apre all'idea del free-text searching e iniziano le prime ricerche nel campo del linguaggio naturale mediante l'utilizzo dell'intelligenza artificiale allo scopo di svolgere il task di question-answering. Lo sviluppo di questi sistemi deve però far

fronte a un limitato quantitativo di dati disponibili, fino a quando negli anni '70 gli sviluppi tecnologici non hanno permesso l'accorpamento di maggiori quantitativi di materiale e la creazione di grandi database computerizzati facilmente accessibili. Nel 1980 un grande numero di database di varia natura sono resi disponibili da servizi online e nel 1990 la diminuzione dei costi e l'aumento della capacità dei dispositivi di archiviazione permette la gestione delle immagini le quali richiedono molto spazio di memoria.

L'information retrieval deve il suo sviluppo a due settori che ne hanno determinato la crescita: in primo luogo lo studio dei metodi analitici di indicizzazione, manuali o automatici e l'utilizzo dell'intelligenza artificiale per l'identificazione dell'informazione e in secondo luogo i metodi statistici e l'implementazione di tecniche probabilistiche utili al raggiungimento del task.

Oggi la facilità di reperire contenuti e la tecnologia opportunamente affinata permettono di estendere la ricerca a contenuti di diversa natura, non solo per il single-media retrieval, ma anche per il cross-media retrieval. Passando da una situazione in cui i contenuti appartenenti al dataset di ricerca hanno la stessa natura della query a uno scenario di documenti eterogenei, il problema in cui ci si imbatte è costituito dal "media gap" [3], ovvero la diversa rappresentazione dei contenuti mediali che non permette una diretta associazione degli stessi.

Questa tesi è incentrata sul video-to-text e sul text-to-video retrieval: partendo da una query in forma di video si desidera recuperare da un dataset gli item di testo coerenti e partendo da una query in forma testuale si desidera ottenere i video corrispondenti.

Oltre al problema del "media gap" illustrato in precedenza si deve far fronte all'ostacolo costituito dall'elevato costo computazionale e di memoria che questo task comporta. Trattandosi di un problema complesso che spesso prevede l'utilizzo di reti neurali per la codifica e l'associazione dei contenuti non è raro che vengano impiegate reti pre-allenate in modo da potersi concentrare su aspetti diversi del processo.

Gli sviluppi nell'ambito dell'intelligenza artificiale, e più precisamente della computer vision e dell'NLP, contribuiscono in modo incisivo al raggiungimento dell'obiettivo.

La computer vision è un campo che utilizza meccanismi di deep learning per estrarre informazioni ed emulare la visione umana. Essa si pone diversi obiettivi tra cui l'image classifiction, l'object detection e l'object tracking.

1. L'image classification consiste nell'assegnare etichette a un'immagine in seguito all'analisi dei pixel che la compongono. La classificazione può essere supervisionata, se l'apprendimento avviene per mezzo di dataset etichettati o non supervisionata nel caso in cui l'apprendimento avvenga in modo automatico.
2. L'object detection individua oggetti facenti parte di specifiche classi all'interno di immagini o video dando informazioni riguardo alla posizione degli stessi. All'interno di un contenuto visivo possono essere individuate più istanze di uno stesso oggetto.
3. L'object tracking consiste nell'identificare e seguire all'interno di un video uno specifico oggetto.

Le applicazioni dei sistemi di computer vision possono spaziare dall'ambito agricolo all'ambito medico oppure possono trovare impiego per gli strumenti di sicurezza, per l'analisi del flusso del traffico o essere integrati in sistemi di guida autonoma.

Il Natural Language Processing riguarda la comprensione, la manipolazione e l'interpretazione del linguaggio da parte delle macchine e tra i task di cui si occupa è possibile evidenziare:

1. Il part-of speech tagging, ovvero l'attribuzione della corretta categoria di parte del discorso in base al contesto.
2. Il word-sense disambiguation, cioé la capacità di dedurre il senso di una parola in base al testo in cui si trova eliminando eventuali ambiguità di significato.

3. Il named entity linking, che consiste nell'associare alle parole chiave di un corpus l'entità corrispondente.

4. Il natural language generation, il quale ha come scopo la produzione di linguaggio naturale a partire da alcuni dati in input.

Le applicazioni dei sistemi di NLP possono includere il filtraggio delle mail, possono essere utili per i sistemi di indirizzamento di pubblicità mirata, per la traduzione automatica dei testi nonché per il riconoscimento del parlato.

L'approccio intrapreso si serve di vari strumenti per il compimento del task e oltre all'impiego di reti pre-allenate, framework e toolkit fa uso anche di knowledge base ovvero banche di dati che contengono informazioni ben strutturate e permettono di derivare le relazioni tra i contenuti. Nello specifico, negli esperimenti effettuati, risultano utili per individuare le relazioni tra le entità derivate da testo e video e sfruttare le proprietà degli item al fine di aumentare l'informazione a disposizione e favorire l'associazione dei contenuti eterogenei coerenti.

Dopo aver analizzato, nei primi capitoli, gli elementi ricorrenti nelle architetture recenti e aver studiato lo stato dell'arte, nel capitolo 4 si approfondiranno gli elementi utilizzati nell'approccio intrapreso in modo da descriverne il funzionamento. Nel capitolo 5 verrà illustrato l'approccio basato sulle knowledge base per lo svolgimento del task di video-to-text retrieval e text-to-video retrieval oggetto di studio di questo elaborato e, infine, verranno discussi i risultati ottenuti dai vari esperimenti intrapresi e verranno proposti ulteriori sviluppi per questa metodologia.

Capitolo 2

Elementi ricorrenti nelle strategie di video e text retrieval

Diversi sono stati i tentativi volti a rendere migliori le prestazioni per ciò che concerne l'associazione di video e testo. La principale difficoltà nel video-to-text retrieval, come nel caso del text-to-video retrieval, è rappresentata dalla diversa natura dei documenti, la quale rende complessa un'associazione diretta dei contenuti.

Come evidenziato in [3], sebbene siano stati fatti tentativi di misura cross-mediale diretta, la maggior parte delle strategie per il calcolo della similarità ricorrono alla creazione di spazi comuni e tra questi si possono evidenziare:

1. I metodi statistici tradizionali che proiettano in modo lineare le caratteristiche visuali e linguistiche e intendono massimizzare la correlazione tra le coppie eterogenee coerenti.

2. I metodi basati sulle deep neural network il cui punto di forza è il livello di astrazione delle informazioni.

3. I metodi basati sui grafi pesati che permettono la gestione delle relazioni intra-media e inter-media.

Prima di procedere con la proiezione della rappresentazione derivante dai dati in input in uno spazio comune, è necessario estrarre tutte le informazioni utili alla comprensione dei contenuti. Gli approcci sono innumerevoli e diversi tra loro, ma si possono individuare alcuni elementi ricorrenti nelle architetture utilizzate che hanno permesso di raggiungere risultati significativi.

2.1 Struttura e funzionamento di una rete neurale

I task di retrieval fanno ampio uso delle reti neurali artificiali, che sono unità fondamentale del deep learning, una branca del machine learning.

Le reti neurali artificiali imitano il funzionamento dei neuroni, le cellule fondamentali del sistema nervoso specializzate nella ricezione, nell'elaborazione e nella trasmissione delle informazioni.

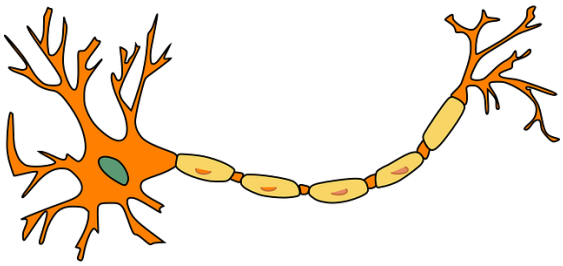

Figura 2.1: Rappresentazione di un neurone

Dall'immagine esemplificativa presente in figura 2.1 è possibile notare che tale cellula è divisa in tre sezioni: il corpo cellulare che contiene il nucleo (in verde nell'immagine) caratterizzato dalla presenza di ramificazioni che prendono il nome di dendriti utili alla ricezione delle informazioni, un prolungamento chiamato assone che trasmette l'impulso elettrico e che termina con un'ulteriore ramificazione che consente il passaggio di informazioni ai dendriti di un'altra cellula nervosa.

Volendo modellizzare il funzionamento di un neurone è necessario pensare all'input come un valore che passa attraverso una serie di nodi caratterizzati da pesi e soglie. I pesi sono dei valori moltiplicativi che indicano l'importanza

dell'informazione in ingresso, mentre le funzioni soglia permettono di determinare se il valore in ingresso è sufficientemente significativo da attivare il neurone.

Come nel caso della cellula del sistema nervoso, nelle unità che costituiscono una rete neurale artificiale si possono individuare tre sezioni: il layer di input, gli hidden layer e il layer di output. L'hidden layer si occupa dell'elaborazione del segnale applicando trasformazioni non lineari in base allo scopo della rete neurale. Tra il layer di input e il layer di output sono interposti uno o più hidden layer (in grigio in figura 2.2). Le connessioni tra i nodi prevedono la possibilità di essere percorsi a croce, ma sempre seguendo la direzione di propagazione del segnale come mostrato in figura 2.2.

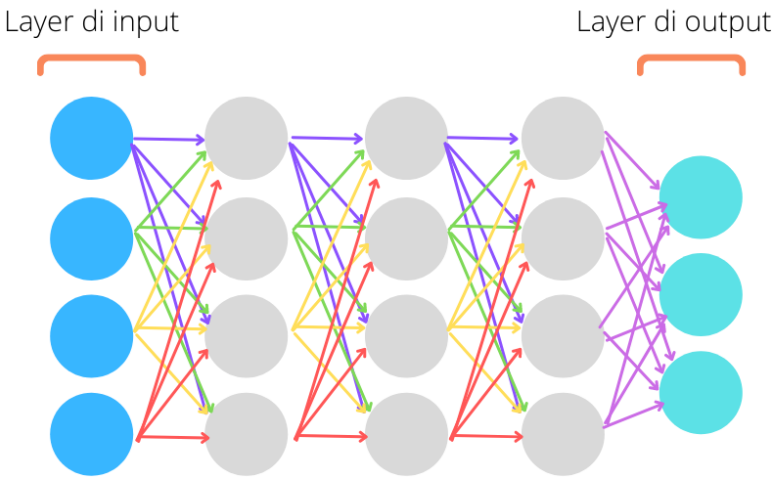

Figura 2.2: Schema di una rete neurale artificiale

Le reti feedforward sono composte da più strati. Le reti multistrato possono apprendere i valori di attivazione tramite la retropropagatione. In fase di training, confrontando i valori di output con un valore di verità viene stimato l'errore commesso dalla rete attraverso la funzione di loss. Quindi, la rete neurale viene

percorsa in senso inverso, a partire dal layer di output verso il layer di input, per l'aggiornamento dei pesi e delle soglie al fine di minimizzare l'errore di predizione. La retropropagazione può essere applicata solo su reti con funzioni di attivazione differenziabili dal momento che la regolazione dei pesi della rete avviene per mezzo di un sistema di ottimizzazione non lineare chiamato gradiente discendente che si basa sulle derivate parziali.

2.2 RNN, LSTM e GRU

Le reti neurali ricorrenti introdotte in [4], RNN [5], sono una classe di reti neurali provviste di memoria a breve termine.

Figura 2.3: Struttura semplificata di una RNN

Come è possibile vedere dalla struttura mostrata in figura 2.3, in corrispondenza dell'hidden state è presente un loop. Tale espediente permette di tenere in considerazione gli output precedenti, i quali contribuiranno alla generazione delle nuove informazioni.

Il problema di questo tipo di rete è rappresentato dal gradiente evanescente. Durante la back propagation, nella fase di aggiornamento dei pesi della rete, i primi layer che hanno elaborato il segnale risentono meno del gradiente, rendendo le correzioni derivanti dai primi output meno significativi. Di fatto è come se la rete si dimenticasse dei primi contributi data la poca influenza del gradiente. Per ovviare a tale problema sono state pensate altre strutture più evolute, in grado di selezionare le informazioni importanti da mantenere in memoria. Questi modelli prendono il nome di LSTM e GRU [6].

La differenza principale rispetto alle RNN è costituita dai gate che permettono di filtrare i contributi degli input determinando quali dimenticare tramite delle funzioni di attivazione. Data, per esempio, una funzione sigmoidea e considerato un valore di input diverso da 0, questo verrà annullato e quindi dimenticato se passando dalla funzione di attivazione essa restituirà valori vicini allo 0, mentre al contrario verrà tenuto in memoria se il valore sarà vicino all'1. L'unità LSTM [7], Long Short-Term Memory unit, è una struttura che sfrutta le funzioni di attivazione al fine di ricordare le informazioni rilevanti.

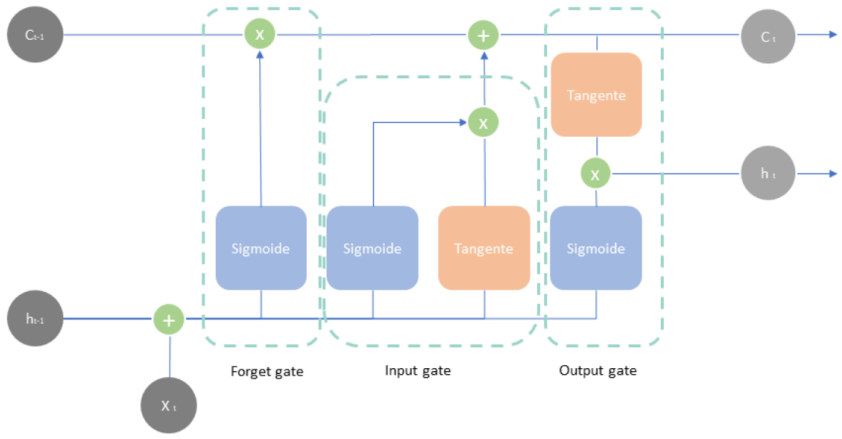

Figura 2.4: Schema di una LSTM Unit [6]

Considerato lo schema in figura 2.4, è possibile individuare tre gate: il forget gate che opera una selezione dei valori derivanti dai passaggi precedenti, l'input gate che aggiorna lo stato corrente definendo il grado d'importanza dell'informazione e l'output gate da cui dipende il valore di hidden state all'ingresso della cella successiva. In ingresso all'unità si hanno il valore di input corrente x_t e il valore dello stato c_{t-1} e dell'hidden state h_{t-1} derivanti dalla cella precedente. Le funzioni sigmoidee introdotte permettono di applicare il principio di memoria selettiva descritto precedentemente, mentre l'introduzione delle funzioni tangente permette di effettuare un riscalamento tale da rendere tutti i valori compresi tra -1 e 1.

Come avvenuto in [8], durante la fase di codifica del testo, l'utilizzo di una

doppia rete LSTM può essere efficace per analizzare in modo più completo le informazioni. Una struttura biLSTM [9] [10], infatti, consiste in una doppia rete che analizza i dati sia in senso corrente sia in senso inverso in modo da beneficiare sia dei contributi derivanti dagli elementi passati sia dagli elementi futuri.

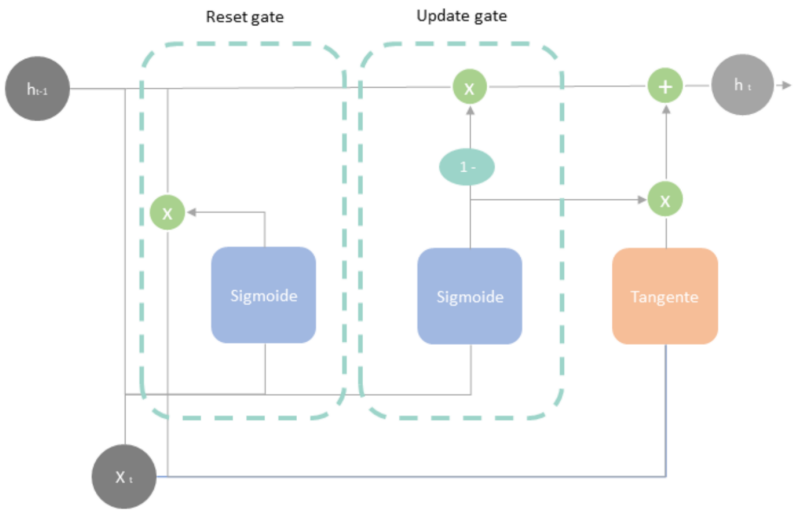

Figura 2.5: Schema di una GRU Unit [6]

Le GRU, Gated Recurrent Unit, introdotte in [11], il cui schema è riportato in figura 2.5, permettono il mantenimento della memoria a lungo termine, ma a differenza delle unità LSTM il flusso delle informazioni è regolato da due gate: il reset gate decide quali delle informazioni precedenti dimenticare, mentre quello di update seleziona le informazioni da tenere e le aggiorna in base ai valori correnti.

Anche nel caso delle GRU è stata creata una struttura biGRU, la quale ha la capacità di dedurre lo stato corrente, prendendo in considerazione sia lo stato precedente, sia lo stato successivo. Nello specifico, la sequenza di input viene analizzata sia in senso normale sia in senso inverso, attraverso una duplice unità composta da forward GRU e backwark GRU. Un esempio di utilizzo delle biGRU nell'ambito di video retrieval è presente in [12]. L'architettura proposta nell'articolo sfrutta questa tipologia di struttura per estrarre dai contenuti visuali e testuali i

pattern temporali.

2.3 Word2Vec

A volte per la codifica del testo viene utilizzata una rete neurale per la rappresentazione vettoriale delle parole che prende il nome di Word2Vec [13]. Questa rete si basa si una struttura in grado di apprendere le relazioni tra le parole costituenti il testo e ne esistono due modelli: il primo, chiamato Continuous Bag-of-Words Model, intende predire la parola corrente in base al contesto, il secondo, il Continuous Skip-gram Model, predice le parole vicine a quella considerata entro una certa finestra di analisi.

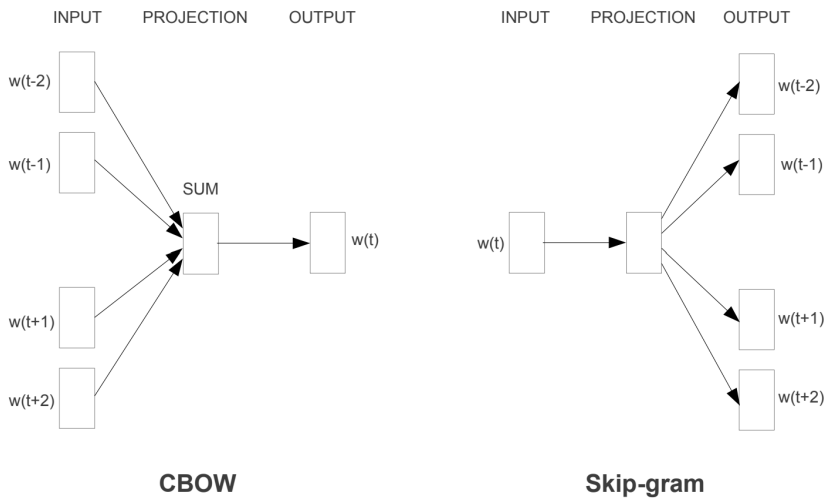

Figura 2.6: Arhitettura dei modelli Continuous Bag-of-Words e Continuous Skip-gram presenti in [13]

L'architettura prevede un layer di input, un hidden layer e un layer di output. Usato un testo come input, questo viene diviso in token e trasformato in un vettore. I vettori derivanti dalle parole costituenti il corpus saranno tali da posizionarsi vicine nello spazio multidimensionale. Questo si verifica dal momento che è probabile che parole simili appartengano a contesti simili.

2.4 CNN

Le reti neurali convoluzionali [14] [15] [16] sono modelli di Deep Learning caratterizzati da una struttura multi-layer gerarchica. I primi livelli sono in grado di riconoscere elementi semplici come linee e curve, mentre i livelli successivi permettono di riconoscere oggetti via via più complessi, ne consegue che tanto più elevato è il numero di blocchi che costituiscono la rete, maggiore sarà la complessità delle caratteristiche estratte.

Presa un'immagine come input, questa può essere vista come una matrice di valori caratterizzanti i pixel. Nel caso in cui si desideri lavorare con un'immagine a colori, sarà necessario separare le matrici dei vari canali costituenti lo spazio colore. L'immagine passerà quindi attraverso tre tipi di layer: il Convolution layer, il layer di Pooling e il Fully Connected layer.

Il CONV è il layer principale della rete neurale. In questa fase viene utilizzato un kernel utile ad effettuare l'operazione di convoluzione sull'immagine in input alla quale viene applicato lo zero padding. Le caratteristiche estratte dipendono dai valori che costituiscono il kernel, che vengono appresi in modo automatico in fase di training. La grandezza della matrice in uscita, invece, dipende dal passo di traslazione del kernel sull'immagine, dal momento che maggiore è il numero di pixel di spostamento del kernel, minore sarà la grandezza della matrice in uscita. Per migliorare le prestazioni del sistema viene aggiunta un'unità lineare rettificata, ReLU, che è definita da una funzione di attivazione non lineare.

Il layer di Pooling effettua il sotto campionamento. Esistono diversi tipi di layer di Pooling. Il max pooling estrae dalla porzione dell'immagine considerata il valore massimo assunto e lo proietta nella mappa sotto campionata, come mostrato in figura 2.7, il mean pooling restituisce in output la media dei valori della porzione considerata e il global avarage pooling media i valori ottenendo in output un array di dimensione unitaria.

Le mappe contenenti informazioni di alto livello, ottenute dall' elaborazione dei livelli precedentemente descritti, vengono trasformate in vettori e quindi posti in ingresso a uno o più layer Fully Connected, FC. Nel livello FC ogni valore di input è connesso a tutti i neuroni che compongono il livello di rete. Il risultato ottenuto

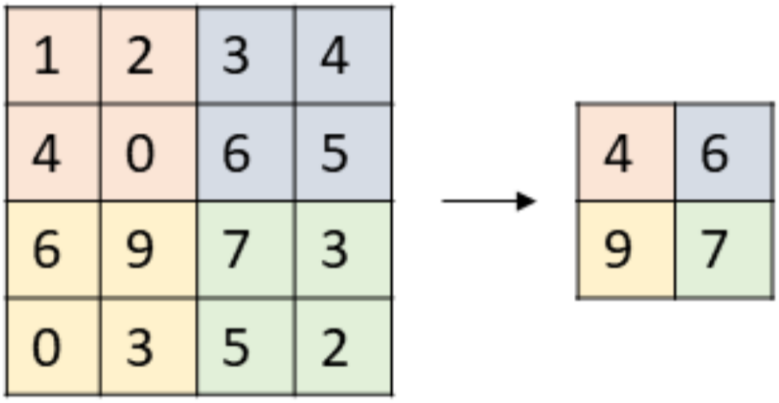

Figura 2.7: Esempio del funzionamento del max pooling

in output viene confrontato, in fase di training, con il valore previsto, la ground truth, attraverso la funzione di loss. Per mezzo del gradiente discendente vengono aggiornati iterativamente i parametri della rete, ad esempio, i pesi del neurone e i valori del kernel, per ottimizzare la predizione.

Sebbene quanto descritto in precedenza faccia riferimento alle immagini è possibile sfruttare il fatto che le CNN lavorino con valori numerici per trasferire il meccanismo sull'analisi del testo. Questo processo di conversione del testo in vettori di valori numerici è reso possibile, per esempio, mediante l'impiego di un modello Word2Vec [13].

Il ruolo dominante delle CNN nel campo della computer vision, le ha rese molto frequenti nelle architetture per il VTR al fine di estrarre le features dalle immagini, come accade in [8]. In [12], invece, le CNN vengono integrate nell'architettura di entrambi i rami di codifica, sia per quanto riguarda il testo sia per quanto riguarda il video.

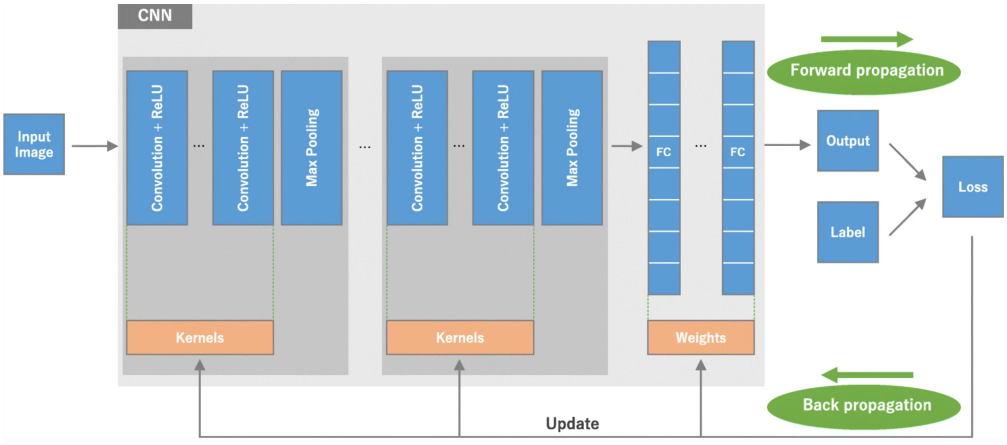

Figura 2.8: Schema di una CNN presente in [14]

2.5 GAN

GAN, Generative Adversarial Networks [17] [18], è un framework composto da una a rete generativa e una rete discriminativa: la prima cerca di produrre un'immagine sintetica a partire da rumore, mentre la seconda ha il compito di riconoscere le immagini reali da quelle generate dalla rete avversaria. La caratteristica di questo framework è la capacità di apprendere usando come unico principio la competizione tra le due componenti. La rete discriminativa, infatti, dà un feedback alla rete generativa dopo aver effettuato un confronto con la ground truth.

Ispirandosi a uno schema presente in [18], in figura 2.9 è stato riportato il funzionamento delle GAN .

Per allenare le reti viene applicata una funzione di minimax, tale da massimizzare la capacità del discriminatore di assegnare la corretta etichetta ai dati di qualunque natura siano e da minimizzare $\log(1 - D(G(\mathbf{z})))$, dove $D(G(\mathbf{z}))$ è l'output del discriminatore derivante dall'analisi dei dati generati. Chiamate $V(G, V)$ la funzione di valore, $p_z(\mathbf{Z})$ le variabili di rumore definite a priori, e $\mathbf{x}$ i dati reali, la formula di training è così definita:

$$max_D min_G V(G, V) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z}))]$$

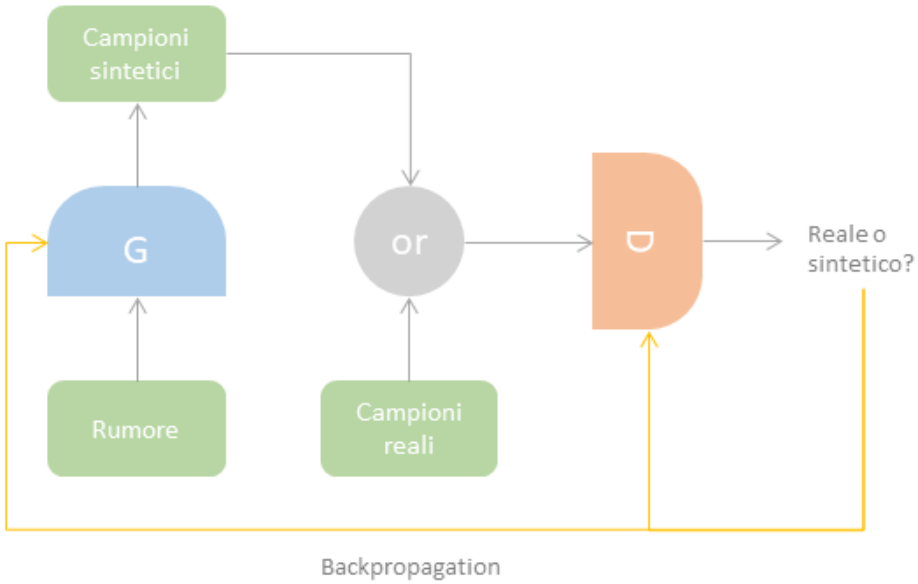

Figura 2.9: Schema di una GAN

Le GAN, nell'ambito del VTR, possono essere usate per allineamento delle feature, nel caso di [8].

2.6 Transformer

I Transformer [19] sono architetture che utilizzano il meccanismo di attenzione come principio di funzionamento delle unità di codifica e decodifica. Codificatore e decodificatore condividono una struttura simile, composta da una pila di sei layer identici tra loro, ma differiscono per il numero di sub-layer: il codificatore possiede un modulo di multi-head attention seguito da una fully connected feed-forward network, FFNN, mentre il decodificatore ha in aggiunta un ulteriore modulo che permette di mantenere la sequenza corretta di analisi. Ogni sub-layer è accompagnato da una residual connection e da un layer di normalizzazione.

La funzione di attenzione mappa il vettore di output a partire tre vettori nominati key, query e value. La vicinanza del vettore di query e key permette di definire i pesi con cui valutare il vettore di value durante la somma pesata descritta dalla

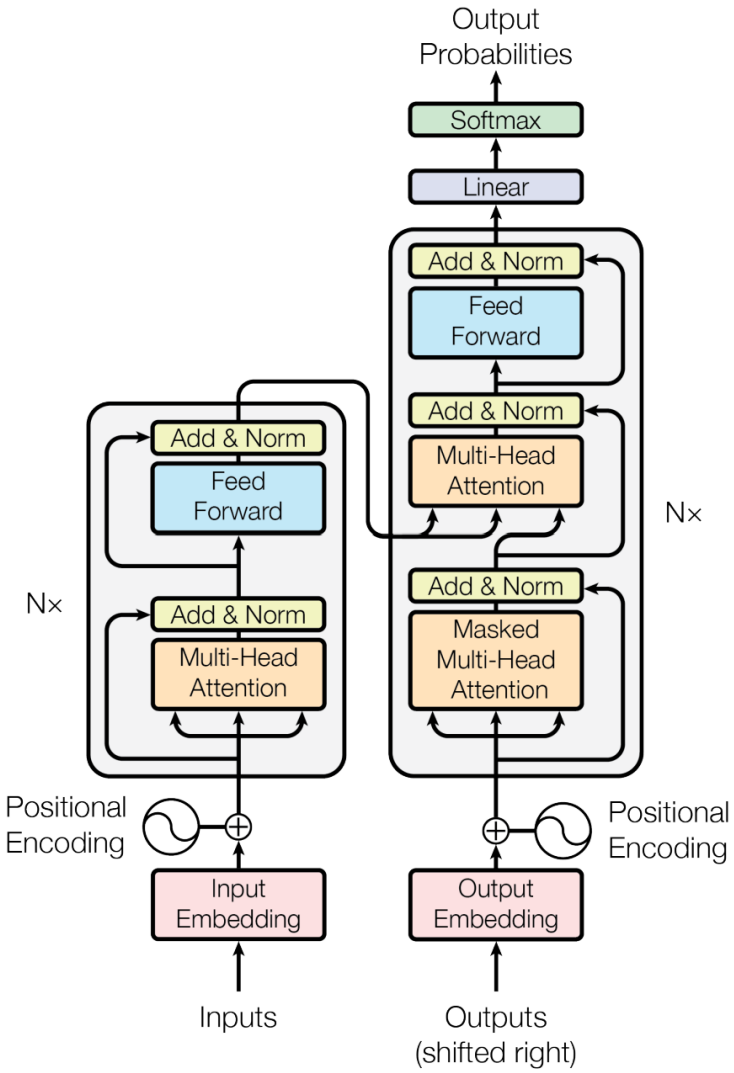

Figura 2.10: Struttura di un Transformer riportata nell'articolo [19]

formula:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Dal momento che vengono analizzate più query contemporaneamente, in calcolo viene effettuato in forma matriciale e nello specifico K, Q e V corrispondono rispettivamente alle matrici contenenti i vettori di key, query e value. Nella formula sopracitata è possibile notare la presenza di un ulteriore parametro, d_k, il quale

rappresenta la dimensione delle query e delle key.

Il meccanismo di multi-head attention permette di avere una rappresentazione più completa delle informazioni mediante la combinazione di diverse proiezioni lineari dei set di query, key e value, secondo la formula:

$$MultiHead(Q, , K, V) = Concat(head_1, ..., head_2)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

dove con W sono indicate le matrici di peso, i cui valori vengono appresi in fase di training.

I Transformer sono stati maggiormente applicati nel campo del NLP mentre sono meno diffusi nell'ambito della computer vision, sebbene di recente siano state effettuate alcune ricerche anche in questo ambito.

2.6.1 BERT

BERT, Bidirectional Encoder Representations from Transformers, introdotto in [20], è un modello multi-layer per il linguaggio naturale sfruttabile per diversi task di NLP. BERT utilizza i Transformer per divincolarsi dalla unidirezionalità di lettura e apprendere informazioni contestuali: tutta la sequenza in input viene letta contemporaneamente così da rendere il modello capace di estrarre informazioni anche sulla base delle parole attigue.

Al fine di essere applicato per diversi utilizzi, BERT è in grado di gestire in ingresso una singola frase o una coppia di frasi. All'inizio dell'input viene posto il token speciale [CLS], mentre il token [SEP] permette di separare le frasi, quindi viene aggiunta l'informazione relativa alla frase di appartenenza. La fase di pre-training di BERT comprende lo svolgimento di due task non supervisionati: il Mask Language Model, MLM, e il Next Sentence Prediction, NSP. Il primo task consiste nel mascherare alcune parole costituenti la frase con un token appropriato, e di far predire al modello il contenuto utilizzando le informazioni provenienti dal contesto. Il secondo risulta utile per l'apprendimento delle relazioni tra le frasi di un testo e consiste nel fornire in input una frase presa da un corpus e di far predire

la successiva. Utilizzando il modello così pre-allenato è possibile raggiungere buoni risultati su diversi task, senza ricorrere ad architetture più complesse e onerose.

Il successo di questo modello si rispecchia nell'ampio utilizzo.

2.6.2 ViT

ViT, introdotto in [21], è un modello scalabile che applica i Transformer per l'image recognition.

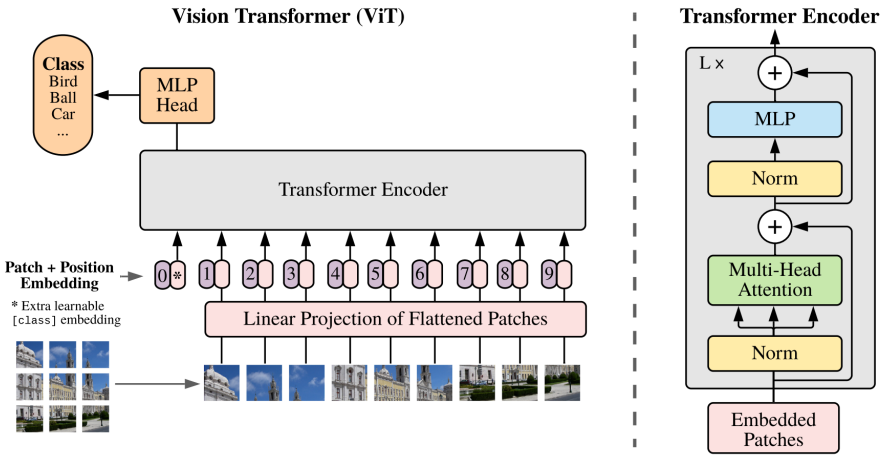

Figura 2.11: Modello di ViT illustrato in [21]

L'immagine utilizzata come input viene segmentata e le porzioni ottenute vengono allineate e proiettate linearmente. Prima di poter giungere al Transformer Encoder, vengono aggiunte le informazioni di posizione. Il codificatore mantiene una struttura simile a quella spiegata precedentemente. Come mostrato in figura 2.11, esso contiene un'alternanza di layer di Multi-head self attention, layer di normalizzazione e Multi-layer Perceptron, MLP, ovvero una classe di FFNN. Il punto di forza di ViT è la capacità di ricavare informazioni derivanti da ogni porzione dell'immagine, dal momento che, sebbene questa sia sezionata, l' applicazione dei Transformer permette un'analisi globale dell'input. I risultati sono significativi a fronte di un ridotto sforzo di pre-training e l'impiego di ViT nel campo del retrieval ha contribuito al raggiungimento di risultati notevoli.

Capitolo 3

Stato dell'arte

Analizzando molteplici articoli riguardanti il text-to-video retrieval e il video-to-text retrieval è evidente che vi siano delle affinità nella scelte architetturali degli elementi scelti. Nella gran parte degli algoritmi che hanno raggiunto lo stato dell'arte di recente, si fa ampio uso dei Transformer in fase di codifica di testo e video e CLIP risulta essere un punto di riferimento. Altre strategie puntano, invece, alla definizione di una nuova pipeline di codifica o al miglioramento degli algoritmi di apprendimento.

Di seguito sono state riportate diverse recenti metodologie dividendole in due gruppi: le strategie che utilizzano CLIP e quelle che non ne fanno uso.

3.1 CLIP

CLIP, introdotto in [22], si pone come obiettivo quello di abbandonare l'utilizzo di classi limitate per l'etichettatura delle immagini in favore di un apprendimento diretto del contenuto di queste a partire dal testo correlato. Nello specifico questa strategia permette di utilizzare il linguaggio naturale per supervisionare l'apprendimento della rappresentazione delle immagini.

Per quanto riguarda l'architettura, per i codificatori di immagini, sono state testate sia le CNN sia i Vision Transformer, mentre per il testo si è ricorso ai Transformer. Partendo da N immagini e N testi vengono prese in considerazione

tutte le N x N possibili coppie e lo scopo di CLIP è quello di apprendere uno spazio multi-modale di embedding tramite il training dei codificatori di testo e di immagini, al fine da massimizzare la similarità cosenica delle coppie corrette.

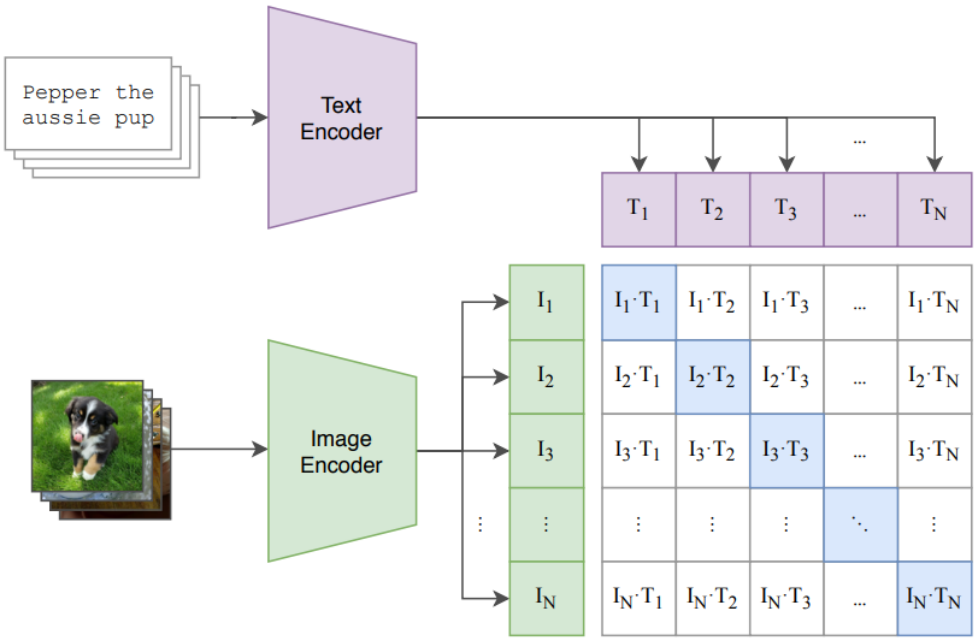

Figura 3.1: Illustrazione del Contrastive pre-training riportata in [22]

Le informazioni estratte dai codificatori, i quali non sono stati inizializzati, vengono proiettate linearmente nello spazio multi-modale di embedding. In fase di test, il codificatore di testo integra le informazioni derivanti dal dataset che si intende utilizzare. Il punto di forza di CLIP non è però solo l'adattabilità ai diversi dataset, ma anche la possibilità di applicazione per diversi task e questo si rispecchia nell'ampia ricorrenza nelle strategie recenti.

3.1.1 CLIP2Video

CLIP2Video [23] è un framewok che intende trasferire la capacità di apprendimento della rappresentazione delle immagini introdotta da CLIP per lo svolgimento del recupero dei video da query testuali.

A partire dalla rappresentazione fornita da CLIP dei contenuti del dataset gli autori hanno incentrato il loro lavoro sulle dipendenze temporali esistenti all'interno dei frame che costituiscono uno stesso video e tra coppie di video e testo. A tale scopo sono stati utilizzati due blocchi: il Temporal Difference Block (TDB) e il Temporal Alignment Block (TAB).

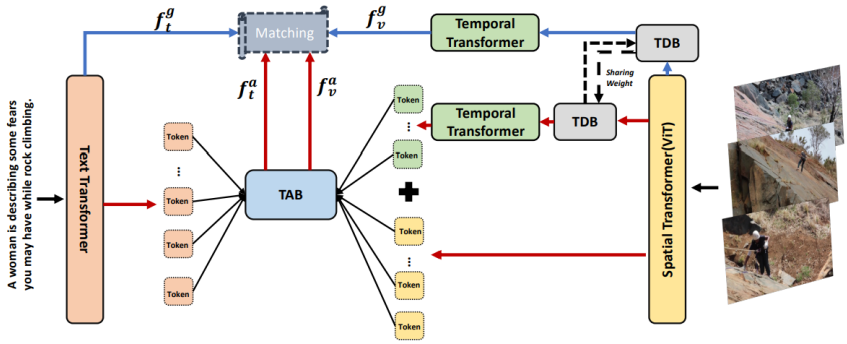

Figura 3.2: Architettura di CLIP2Video presentata in [23]

A partire dalle caratteristiche estratte dal video tramite ViT, il quale non tiene conto delle informazioni temporali, il blocco TDB permette di individuare i cambiamenti nell'azione analizzando frame adiacenti. Con questi dati, il Transformer temporale è in grado di effettuare una migliore codifica delle informazioni legate al movimento. I token in uscita dal blocco temporale e i token provenienti dal testo vengono allineati nel TAB nel quale le rappresentazioni derivanti dalle due modalità vengono aggregate attorno a dei centri in base al contenuto. Successivamente, le azioni individuate dai frame e le azioni descritte dal testo vengono utilizzate per enfatizzare gli elementi contestuali e aggiustare la posizione dei centri.

3.1.2 CAMoE

Dal momento che CLIP è stato allenato su un vasto dataset i modelli che lo sfruttano sono propensi a generare overfitting in fase di training.

Gli autori di [24] propongono un nuovo metodo chiamato CAMoE. Il nome deriva dalla composizione dei termini: multi-stream Corpus Alignment (CA) e

Mixture-of-Expert (MoE).

Il testo viene codificato utilizzando BERT pre-allenato con CLIP, mentre il video viene codificato con ViT anch'esso pre-allenato con CLIP.

L'architettura prevede di sfruttare degli esperti in grado di apprendere in modo separato rappresentazioni di entità e azioni. Per quanto riguarda l'analisi del testo vengono utilizzati dei modelli di part-of-speech tagging e strategie di sentence generation (SGS) per il riconoscimento di verbi e nomi. In fase di analisi del video, come è possibile vedere dalla Figura 3.3, il flusso delle informazioni provenienti da Entity Expert e Action Expert è regolato da un gate sulla base di un punteggio di importanza, in modo da integrare e rafforzare la rappresentazione del Fusion Expert.

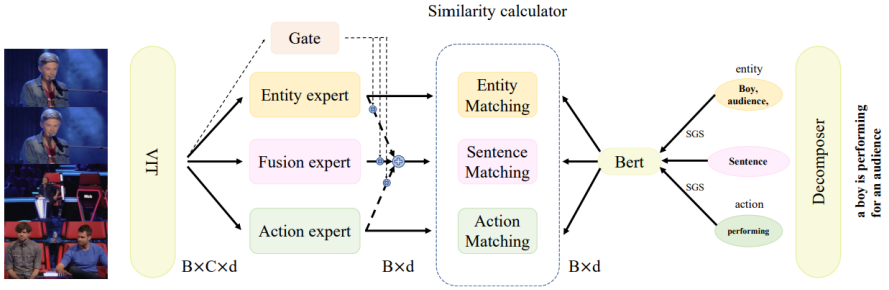

Figura 3.3: Architettura di CAMoE riportata in [24]

A questo punto, le informazioni derivanti da testo e video sono paragonabili e ne viene calcolata la similarità.

Il secondo contributo di questo lavoro consiste nella definizione della Dual Softmax loss. Nei lavori precedenti il Softmax risulta applicato alla singola operazione di recupero rendendo il risultato non ottimale. Uno stesso testo può essere, infatti, associabile a più video a causa della minore quantità di informazione determinata dalla sua natura. Si è evidenziato che comparando i punteggi di similarità di un testo con tutti i video e viceversa i risultati ne beneficiano.

Partendo dalla definizione di similarità cosenica:

$$sim(v_i, s_i) = \frac{v_i \cdot s_i}{||v_i|| \cdot ||s_i||}$$

e dalla definizione delle prior matrix per il video-to-text e il text-to-video retrieval:

$$Pr_{i,j}^{v2t} = \frac{\exp\left(l \cdot sim(v_i, s_i)\right)}{\sum_{j=1}^{B} \exp\left(l \cdot sim(v_j, s_i)\right)}$$

$$Pr_{i,j}^{t2v} = \frac{\exp\left(l \cdot sim(v_i, s_i)\right)}{\sum_{j=1}^{B} \exp\left(l \cdot sim(v_i, s_j)\right)}$$

dove t rappresenta il sentence, entity e action matching, ı e ȷ sono gli indici del batch e ł è un parametro di scala, le funzioni di loss sono così definite

$$L_t^{v2t} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp\left(l \cdot sim(v_i, s_i) \cdot Pr_{i,i}^{v2t}\right)}{\sum_{j=1}^{B} \exp\left(l \cdot sim(v_i, s_j) \cdot Pr_{i,j}^{v2t}\right)}$$

$$L_t^{t2v} = -\frac{1}{B} \sum_{i}^{B} \log \frac{\exp\left(l \cdot sim(v_i, s_i) \cdot Pr_{i,i}^{t2v}\right)}{\sum_{j=1}^{B} \exp\left(l \cdot sim(v_j, s_i) \cdot Pr_{j,i}^{t2v}\right)}$$

Le sperimentazioni hanno permesso di evidenziare un miglioramento dei risultati attribuibili all'aggiunta della prior matrix calcolata diagonalmente.

3.1.3 CLIP2TV

CLIP2TV [25] prevede l'utilizzo di CLIP per la codifica dei contenuti nelle due modalità, ma al Visual Transformer viene associato un Temporal Transformer per evidenziare le informazioni temporali. Si procede, quindi, alla proiezione dei frame di embedding e dei token di embedding in un sottospazio multi dimensionale dove si cerca di allineare i video e le caption utilizzando la similarità cosenica e il contrastive loss.

Dal momento che difficilmente vi è una completa corrispondenza tra contenuti di natura eterogenea, viene utilizzato il Momentum Distillation. Si tratta di un'unità

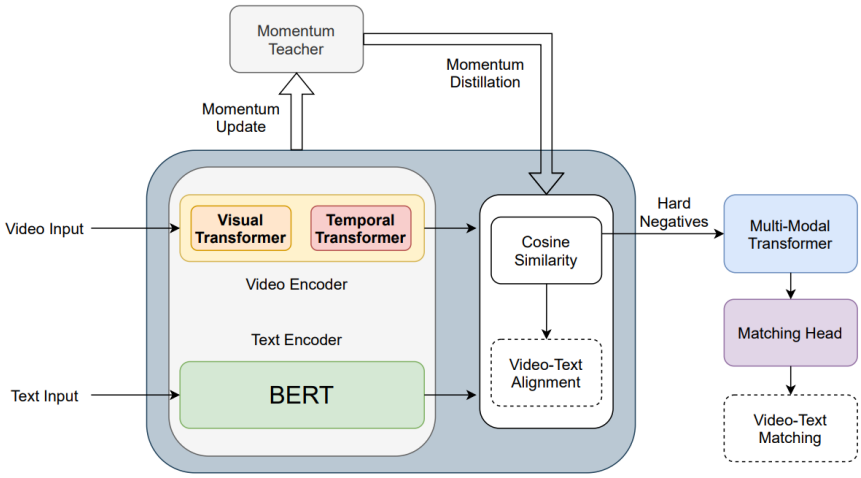

Figura 3.4: Architettura di CLIP2TV riportata in [25]

che in fase di training influisce sul calcolo della similarità e del contrastive loss[1] tramite l'exponential moving average, ovvero una media che permette di sfruttare le informazioni passate per creare una linea di tendenza che dà maggior rilevanza alle informazioni più recenti. Vengono, infine, estratti solo i campioni fortemente negativi che entrano nell'unità di fusione cross-modale. Questa architettura e l'utilizzo di una funzione Dual Softmax simmetrica in fase di inferenza, permette di distinguere tra elementi che hanno caratteristiche simili, ma semantica diversa.

3.2 Altri metodi

3.2.1 Codificatore duale e spazio ibrido

Allo scopo di svolgere il task di text-to-video retrieval [12] crea un'architettura in grado di codificare separatamente video e testo e sfrutta uno spazio ibrido che

[1]Il contrastive loss, introdotto in [26], è una funzione di loss che ha la caratterista di apprendere i parametri di peso in modo tale da avvicinare gli elementi simili (campioni positivi) di allontanare gli elementi diversi (campioni negativi).

unisce la buona interpretabilità dello spazio di concetto e le elevate prestazioni dello spazio latente.

La pipeline di codifica è simile per video e testo:

1. Il mean pooling viene utilizzato per estrarre i pattern globali effettuando una media delle feature estratte dai frame e degli one-hot vector generati dal testo.

2. Le biGRU sono utilizzate per l'estrazione dei pattern temporali.

3. Una biGRU-CNN, formata da una rete convoluzionale posta al di sopra di una biGRU, viene impiegata per evidenziare i pattern locali.

Si procede, dunque, proiettando linearmente le codifiche nello spazio latente e nello spazio di concetto. Lo spazio latente mira a rappresentare nello spazio i campioni in base alle posizioni relative per cui concetti distanti tra loro possono avere i centri vicini. Al contrario, ogni dimensione della seconda tipologia di spazio rappresenta un concetto, come può essere quello di "ragazzo" o "spiaggia" e quindi è direttamente interpretabile. La combinazione dei due permette di prendere gli aspetti positivi di entrambi e ottenere risultati migliori.

3.2.2 MAN

MAN, Multi-level Alignment Network, è il metodo di allineamento cross-modale proposto in [8]. La problematica principale che gli autori intendono risolvere è costituita dal diverso dominio dei dataset, utilizzati in fase di training e in fase di testing, i quali influiscono negativamente sui risultati. Per risolvere tale problema viene proposto un metodo di allineamento nello spazio comune volto a superare i gap semantici, di dominio e di modalità, utilizzando un metodo di adattamento non supervisionato, ovvero in fase di training non si ha accesso ai dati etichettati di testing.

L'architettura prevede l'utilizzo di CNN pre-allenate per la codifica delle immagini, mentre per il testo sono impiegate unità biLSTM e layer di pooling. Le caratteristiche estratte passano in un layer FC e vengono proiettate nello spazio comune dove avviene il triplice allineamento:

1. L'allineamento semantico permette di avvicinare le coppie di video e di testo in base alla rilevanza semantica.

2. L'allineamento di cross-dominio permette di apprendere le caratteristiche indipendentemente dal fatto che si tratti di un dataset di training o di testing e ciò è possibile grazie all'impiego delle GAN.

3. L'allineamento cross-modale permette di agire per ridurre la differente distribuzione e rappresentazione dovute alla natura diversa dei contenuti da associare.

L'unione di questi questi tre allineamenti ha permesso di ottenere una maggior generalizzazione dei dati di target.

3.2.3 TACo

TACo [27], Token-Aware Cascade contrastive learning è un algoritmo per il text-to-video retrieval. Il codificatore video è composto da layer di self-attention e l'estrazione delle feature si suppone essere stata effettuta con modelli di CNN pre-allenati, mentre per il codificatore testuale viene impiegato BERT opportunamente allenato. In seguito, è presente un modulo di fusione multi-modale che riceve in input le caratteristiche provenienti da testo e video, ai quali vengono aggiunti dei token per facilitare il riconoscimento della natura dei dati. Il modello viene addestrato con la funzione token-aware cascade contrastive loss. Questa funzione si concentra su un sottoinsieme che comprende le parole di significato, come nomi e verbi, dal momento che queste risultano essere più facilmente allineabili ai contenuti visivi rispetto alle parole funzionali. Per aumentare l'efficienza, i layer di fusione multi-modale vengono allenati mediante un metodo di campionamento in cascata che permette la selezione di alcuni campioni hard-negative. Il calcolo dei punteggi di allineamento delle coppie di video e testo effettuato prima del blocco multi-modale porta a un apprendimento più efficiente dei layer che lo costituiscono.

Capitolo 4

Elementi fondamentali per l'approccio intrapreso

Dopo aver analizzato alcune delle strategie più recenti per raggiungere il task del video-to-text retrieval e del text-to-video retrieval, si intende presentare gli elementi utili alla metodologia intrapresa.

4.1 Object detection

4.1.1 Definizione e panoramica

Come riportato in [28] l'operazione di object detection permette il riconoscimento di istanze di alcune specifiche classi all'interno di un'immagine. Gli oggetti però possono occupare diverse aree, avere diverse grandezze, diversi orientamenti o differire per caratteristiche qualitative quali, ad esempio, colore e illuminazione e variare la forma. Si pensi per esempio, di voler riconoscere tutti i volti presenti in una serie di fotografie: potrebbe essere raffigurata una sola persona in primo piano, o molteplici individui a diverse distanze nello spazio ripreso e i volti differirebbero per posizione e colore degli occhi, forma del naso e della bocca e così via. Anche lo sfondo contribuisce a determinare la facilità del riconoscimento. Un oggetto su sfondo neutro e in posizione centrale sarà più facilmente riconoscibile rispetto a uno

inserito in un contesto più complesso e inoltre gli oggetti semi-occlusi o le ombre possono inficiare sul risultato così come le distorsioni prospettiche. Si consideri per esempio la figura 4.1 raffigurante un cucciolo di cane rannicchiato in una ciotola. Questa fotografia rappresenta diverse complessità di riconoscimento della classe. Il soggetto si trova in una posizione non usuale ed è parzialmente coperto dal bordo della ciotola. Il sistema di object detection non riesce a individuare correttamente il tipo di animale e ritiene che si tratti di un gatto con un valore di accuratezza di del 99%.

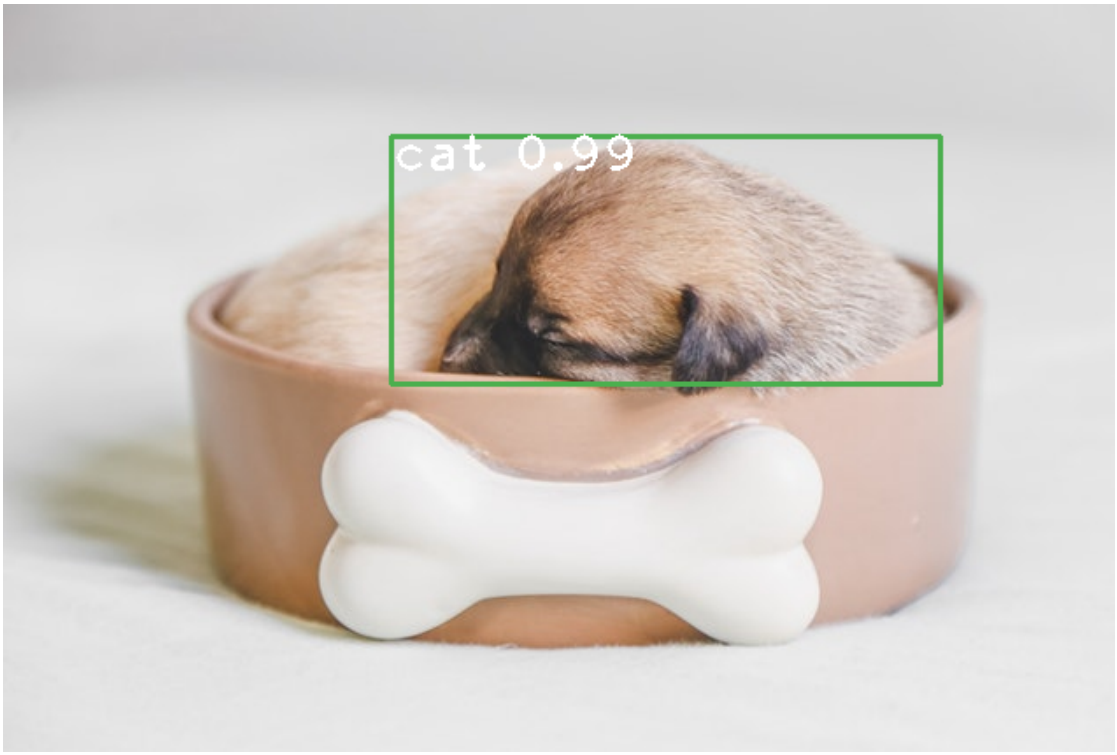

Figura 4.1: Immagine con soggetto semi-occluso e in posizione non ottimale

La molteplicità delle casistiche in cui è possibile si presentino gli oggetti rende necessario creare un modello che generalizzi le caratteristiche delle istanze proprie di una determinata classe. Oltre alla strategia di analisi delle immagini, un altro aspetto da tenere in considerazione è la categoria di object detection utilizzata. Esistono due modelli fondamentali per eseguire questo task: il metodo generativo modella la distribuzione dei dati data l'etichetta mentre e il metodo discriminativo restituisce

la probabilità dell'etichetta avendo fornito il dato. Il livello di generalizzazione dipende non solo dal modello, ma anche dal dataset utilizzato e dalla quantità di campioni a disposizione. Più vasto è il range di item analizzati, maggiore sarà la variabilità degli elementi appresi. È possibile integrare le informazioni così raccolte con opportune funzioni locali e caratteristiche invarianti alle trasformazioni, alle posizioni e all'orientamento, nonché attraverso variabili latenti che parametrizzano la variabilità restituendo valori non direttamente osservabili e interpretabili.

Per riconoscere la presenza di un oggetto all'interno dell'immagine, questa viene analizzata su tutta l'area, generalmente utilizzando più scale e diverse risoluzioni. Per rendere meno onerosi i calcoli esistono alcuni espedienti tra i quali:

1. l'attenzione selettiva su aree che circondano una specifica caratteristica, alla ricerca di ulteriori punti di riferimento che permettano di confermare la presenza uno specifico oggetto

2. l'utilizzo di classificatori in cascata che discriminino il contenuto delle finestre di analisi in modo sempre più preciso, diminuendo il numero di falsi positivi riconosciuti man mano che si procede nella sequenza.

3. i sistemi di decomposizione e l'eliminazione sempre più fine delle porzioni di sfondo man mano che ci si avvicina all'oggetto.

Il rilevamento degli oggetti è uno dei task fondamentali del machine learning e i progressi tecnologici che hanno reso disponibili GPU sempre più potenti hanno reso possibili notevoli miglioramenti in questo campo.

Le tecniche recentemente proposte si dividono in sistemi di object detection a uno stadio e sistemi di object detection a due stadi. I sistemi di rilevamento a due stadi sono sistemi più lenti perché prevedono una ricerca preliminare della regione di interesse che viene proposta per la classificazione. I sistemi uno stadio eliminano lo step di proposta della regione e sono nettamente più veloci a scapito di una minore precisione nel riconoscimento degli oggetti di forma irregolare. Fanno parte delle architetture a uno stadio i sistemi RetinaNet e YOLO con le sue varianti.

4.1.2 RentinaNet

RetinaNet [29] è un'architettura utile per effettuare object detection e fa parte dei sistemi a uno stadio. Essa è costituita da una backbone e due sotto-reti, una per la classificazione degli oggetti e una per definire con precisione la posizione dei bounding box.

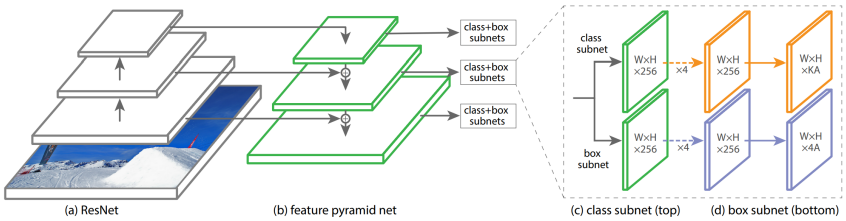

Figura 4.2: Architettura di una RetinaNet introdotta in

Una RetinaNet è formata da una backbone costituita da una Feature Piramyde Network che permette l'analisi di un'immagine estendendo l'informazione derivante dalla CNN su cui è costruita per effettuare un'analisi multi-scala. In figura 4.2 la struttura piramidale è costruita al di sopra di una ResNet, una tipologia di CNN caratterizzata dalla presenza di percorsi alternativi per ovviare al problema della saturazione del gradiente dovuta ai numerosi layer che la compongono, permettendole di saltarne alcuni e mantenendo l'informazione più grezza.
Sulla backbone sono state inserite due sottoreti: la sottorete creata per la classificazione predice il contenuto della porzione di immagine interna ad ogni anchor box, i quali sono invarianti per traslazione, sulla base delle K classi disponibili, mentre la seconda sottorete apprende le informazioni di locazione predicendo gli offset tra punti di riferimento e il box di gound truth.

Questo design permette la creazione di un rilevatore a uno stadio fortemente efficiente e decisamente veloce.

4.1.3 YOLO

YOLO, introdotto in [30], è un sistema per l'object detection veloce al punto da riuscire a processare le immagini in real-time, a 45 frames al secondo, mantenendo elevate prestazioni.

Rispetto ad altri sistemi precedenti caratterizzati da pipeline di classificazione più complesse basate, ad esempio, su finestre a scorrimento o di architetture che necessitano di allenare e mettere a punto ogni componente singolarmente, YOLO è un modello più snello che analizza l'immagine globalmente riuscendo a codificare informazioni contestuali. Ciò avviene tramite una griglia che divide l'immagine in celle ognuna delle quali è responsabile della rilevazione dell'oggetto al suo interno e di predirne B bounding box di cui si definiscono le dimensioni, la posizione dei centri e dei relativi punteggi di confidenza.

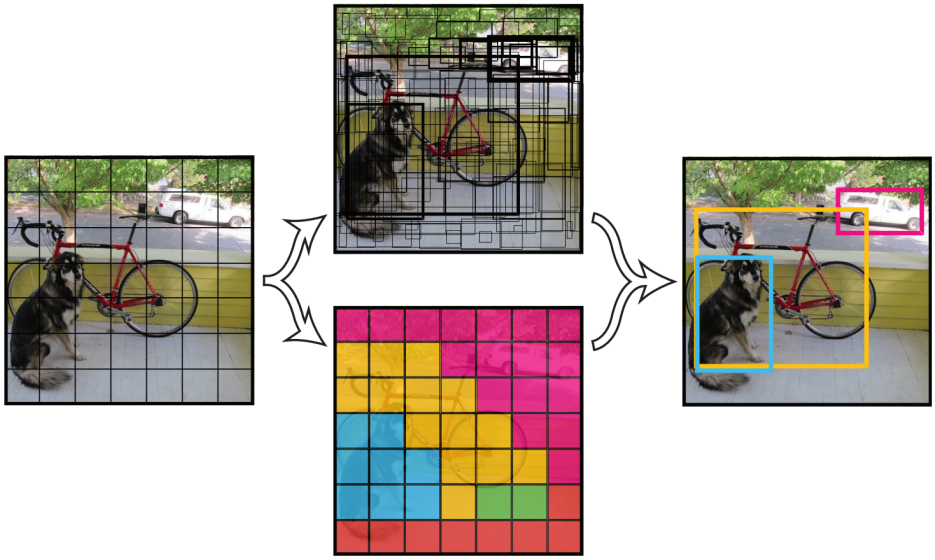

Figura 4.3: Modello di YOLO riportata in [30]

La probabilità che l'oggetto riconosciuto sia effettivamente quello presente è calcolata come segue: $Pr(Object) * IOU_{pred}^{truth}$, dove IOU_{pred}^{truth} indica l'intersection over union tra la ground truth e il box predetto.

In fase di testing i punteggi tengono conto sia del livello di confidenza nella predizione sia della precisione con cui il box individua l'oggetto:

$$Pr(class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

A tal fine, YOLO utilizza un'unica rete neurale composta da 24 layer convoluzionali per l'estrazione delle feature e 2 layer FC per predire coordinate e probabilità.

Una delle caratteristiche di YOLO è di necessitare di una sola valutazione sia in fase di training sia in fase di inferenza e in entrambi i casi è stato utilizzata darknet [31], una rete open source.

In [32] sono stati apportati alcuni miglioramenti al modello. Diversamente da quanto accadeva in YOLO, YOLOv3 utilizza un metodo automatico per la ricerca delle prior matrix ottimali, introdotto in una versione precedente [33], che sfrutta il k-means clustering e utilizza questi cluster dimensionali come anchor box, ovvero aree rettangolari specializzate per il riconoscimento di specifici oggetti. La predizione del box avviene a partire da quattro parametri calcolati dalla rete t_x, t_y, t_w e t_h tramite le seguenti relazioni:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

dove c_x, c_y è il centro dell'immagine corrispondente all'angolo in alto a destra, σ è la funzione sigmoidea e p_w e p_h sono le dimensioni del bounding box prior. Anche l'architettura ha subito delle modifiche dal momento che sono stati aggiunti diversi layer convoluzionali e residual connection le quali consentono di saltare alcuni livelli dell'architettura mantenendo intatta l'informazione acquisita precedentemente.

Sebbene l'architettura sia più complessa le prestazioni in termini di precisione e di velocità rimangono elevate.

4.2 Wikidata

4.2.1 Descrizione generale

Come riportato in [34], Wikidata [1] è una knowledge base collaborativa e multilingue creata nell'ottobre del 2012 da Wikimedia allo scopo di facilitare l'accesso e la gestione dei dati di Wikipedia su scala globale. L'informazione strutturata permette l'utilizzo e l'elaborazione dei dati da parte di software e applicazioni che possono comprenderli e sfruttarli per molteplici scopi.

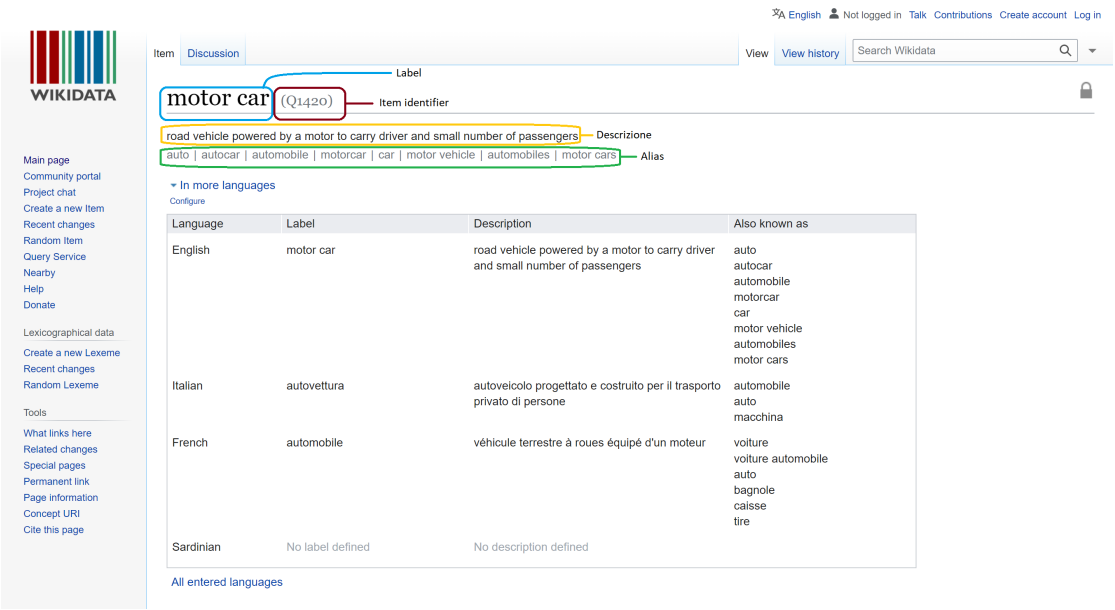

Figura 4.4: Esempio di item di Wikidata

Il mantenimento, l'integrazione e la creazione dei contenuti avvengono in una modalità ibrida che beneficia sia dell'apporto della community sia dell'impiego di sistemi automatici. Questo database è modificabile da qualsiasi utente e data la molteplicità dei collaboratori è prevista la coesistenza di elementi plurimi fornendo un meccanismo di organizzazione dei dati conflittuali. Al fine di ampliare l'informazione e supportare i dati riportati è possibile indicare i riferimenti alle fonti,

[1] `https://www.wikidata.org/`

aggiungere citazioni e creare collegamenti con altri database.

Come illustrato in figura 4.4, ogni item di Wikidata è caratterizzato da un label e un termine identificativo univoco formato dalla lettera **Q** seguita da alcune cifre. È possibile, inoltre, trovare una descrizione e un numero non specificato di alias.

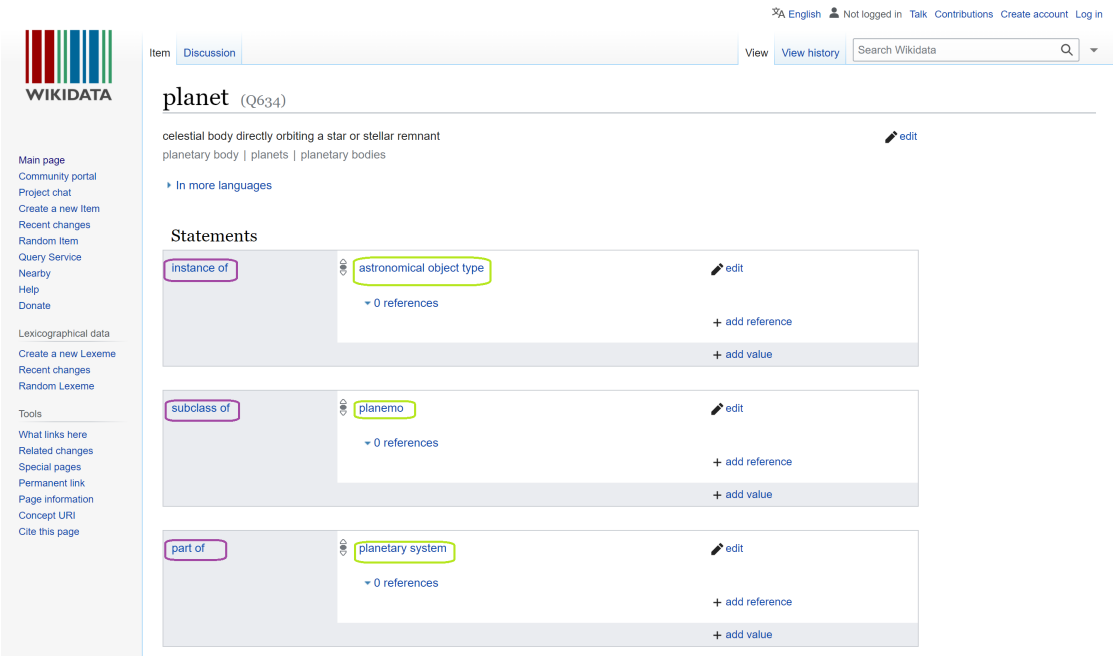

Figura 4.5: Screenshot esemplificativo di uno schema contenente alcune delle proprietà di un item di Wikidata

Come è possibile notare dalla figura 4.5, nella pagina relativa ad un item si può trovare la tabella "Statements" che contiene le proprietà di quell'elemento (evidenziate da un riquadro viola nell'immagine) e i rispettivi valori associati (indicati dal contorno verde).

Wikidata accetta molteplici datatype che comprendono [2]:

1. file appartenenti ai media più comuni
2. coordinate geografiche

[2] `https://www.wikidata.org/wiki/Help:Data_type`

3. item
4. proprietà
5. stringhe
6. testi monolingue
7. identificatori esterni
8. quantità
9. date e ore
10. URL
11. espressioni matematiche
12. tabelle
13. lessemi
14. cartine geografiche
15. notazioni musicali
16. moduli o schemi

Nel caso in cui il valore di una proprietà non sia determinabile, è possibile specificarlo tramite l'apposita dicitura *unknown value* o *no value*. Ciò risulta utile nel caso in cui l'informazione non sia oggettivamente reperibile al fine di rendere l'insieme delle caratteristiche completo.

4.2.2 Interrogare Wikidata

Wikidata mette a disposizione le proprie informazioni e contenuti che possono essere utili per integrare software e applicazioni di vario genere.

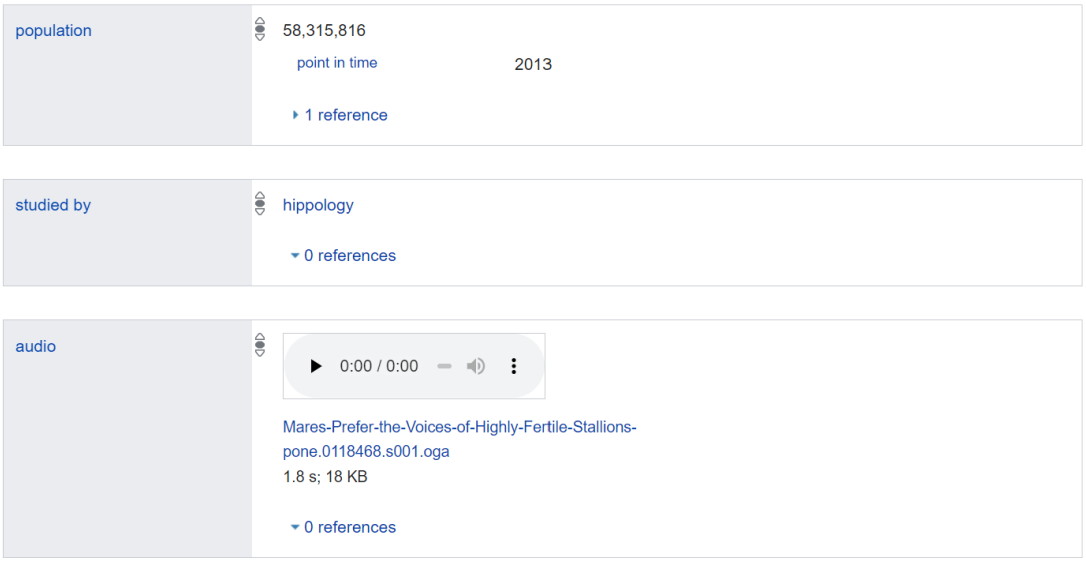

Figura 4.6: Esempio di valori appartenenti a datatype diversi per l'item "horse"

Uno dei punti di forza di questa knowledge base è la possibilità di essere indipendente dalla lingua scelta e ciò è possibile grazie all'utilizzo degli ID i quali non sono vincolati alle label associate.

Per accedere alle informazioni è possibile utilizzare opportune query le quali utilizzano SPARQL come linguaggio di interrogazione.

Il database può essere visto come un insieme di triplette contenti item|proprietà|valore|. Item e valore sono caratterizzati da un ID che inizia con la lettera **Q** mentre quello delle proprietà inizia con la lettera **P**.

Utilizzando il tool "Wikidata Query Service [3]" è possibile interrogare il database. Si supponga di voler ricavare tutti i valori relativi alla proprietà "color" (P462) dell'item "apple" (Q89) e i relativi label in italiano e in inglese.

La query effettuata restituisce due valori: il primo corrisponde agli ID dei colori associati alla proprietà in esame, la seconda le label corrispondenti, come è possibile vedere in figura 4.7.

[3] `https://query.wikidata.org/`

```
SELECT ?colors ?colorsLabel
WHERE {
  wd:Q89 wdt:P462 ?colors.
  ?colors rdfs:label ?colorsLabel
  FILTER(LANG(?colorsLabel) = 'it' || LANG(?colorsLabel) = 'en')
}
```

8 risultati in 635 ms Codice Scarica Collegamento

colors	colorsLabel
wd:Q943	yellow
wd:Q943	giallo
wd:Q3133	green
wd:Q3133	verde
wd:Q3142	rosso
wd:Q3142	red
wd:Q429220	pink
wd:Q429220	rosa

Figura 4.7: Screenshot dei risultati ottenuti dalla query di esempio

Cercando nella pagina relativa all'item "apple", nella tabella della proprietà "color" presentata in figura 4.8, è possibile osservare la corrispondenza dei valori ottenuti.

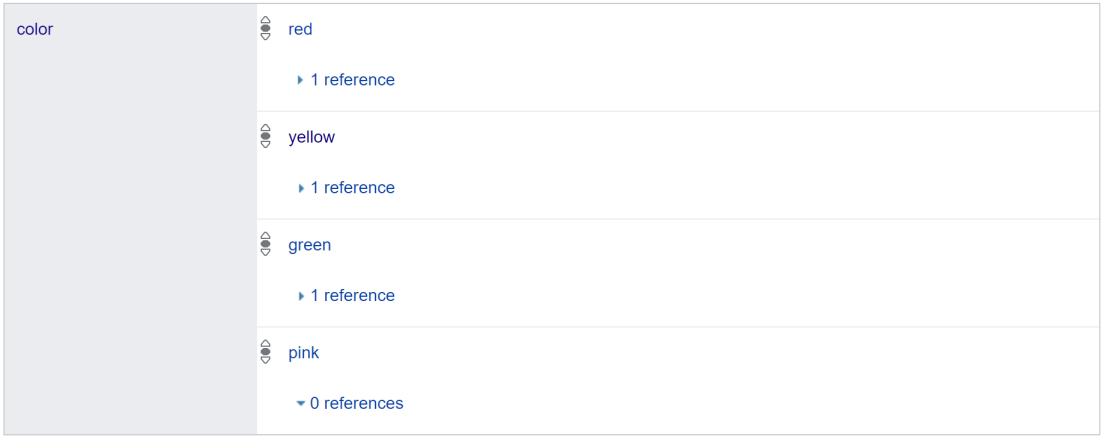

Figura 4.8: Screenshot della tabella "color" dell'item "apple"

Sia le proprietà sia gli item sono entità e sono associati alla relativa URI

caratterizzata dalla forma *http://www.wikidata.org/entity/ID*. Ogni entità è univoca sebbene ad essa possano essere associati molteplici label nel caso in cui siano presenti alias o lingue diverse.

4.3 API di Google per l'estrazione delle entità

Uno strumento molto utile per le elaborazioni sono le API di Google [4]. Google Cloud mette a disposizione molteplici strumenti utili al Cloud computing, all'analisi dei dati e al machine learning utilizzabili sfruttando diversi linguaggi di programmazione quali C#, Go, Java, Node.js, PHP, Python e Ruby e permettendo un approccio guidato all'utilizzo. Tra gli strumenti per il machine learning sono presenti le API Vision e le API Cloud Natural Language: il primo permette di classificare le immagini in base al contenuto, di trovare eventuali testi e può riconoscere le emozioni, il secondo analizza il contenuto del testo per diversi task tra i quali l'estrazione delle entità e l'analisi sintattica.

Uno dei vantaggi dell'utilizzo di tale strumento è il grande quantitativo di modelli pre-allenati. La possibilità di legare i contenuti estratti dai media ai link relativi alle entità di Wikidata contribuisce a renderla una soluzione utile per il raggiungimento dell'obiettivo prefissato in questo elaborato.

4.4 Babelfy e TextRazor

Babelfy[5] è un software utile a svolgere il task di disambiguation a partire da un testo. Esso si basa sulla rete semantica BabelNet che estrae le entità, cerca i possibili significati e tramite un sistema di grafi permette di riconoscere il più affine.

TextRazor[6] è un'infrastruttura per l'analisi del testo che consente di intraprendere alcuni task tra cui l'entity extraction e l'entity linking.

[4] `https://cloud.google.com/apis/`

[5] http://babelfy.org/

[6] https://www.textrazor.com/

4.5 NLTK e WordNet

NLTK [35] è un toolkit di linguaggio naturale gratuito e open source che permette di eseguire diversi task utilizzando Python come linguaggio di programmazione. NLTK mette a disposizione diverse librerie e corpora come ad esempio WordNet [36].

WordNet è un database lessicale caratterizzato da un'organizzazione dei termini in synset. Questi gruppi di sinonimi sono legati tra loro in base a diverse relazioni di parentela che permettono, ad esempio, di associare termini più specifici a set dal significato più generico creando una struttura ad albero. All'interno del database è possibile riconoscere due tipologie di parole: i tipi, ovvero i nomi comuni e le istanze, ovvero nomi specifici che costituiscono le foglie dell'albero. Questa struttura gerarchica è ripresa anche nel contesto dei verbi che sono legati sulla base della specificità dell'azione descritta, mentre gli aggettivi, oltre che ai termini di significato simile, vengono associati ai contrari. WordNet non collega, però, solo termini che appartengono alla stessa parte del discorso, ma tiene conto dei legami tra verbi, aggettivi, nomi e avverbi che condividono una radice comune o un significato affine.

La suite NLKT permette di risolvere diversi task nell'ambito dell'NLP come ad esempio la tokenization, la il POS tagging o la lemmatization, utilizzati nella metodologia intrapresa.

Tokenization [37] consiste nel dividere un testo negli elementi che lo compongono. A partite da un corpus è possibile desiderare di ottenere le frasi o le parole che lo costituiscono. Questo task risulta semplice per alcune lingue come ad esempio l'inglese o l'italiano, in cui le singole parole sono separate da spazi, mentre in altri casi si rende necessario utilizzare metodi diversi che richiedono l'utilizzo di dizionari o l'apprendimento di regole grammaticali.

POS tagging fa riferimento all'etichettatura delle parti del discorso (**P**art **O**f **S**peech). Una frase è composta da termini che fanno parte di classi morfologiche

diverse come ad esempio nomi, aggettivi, avverbi, verbi, articoli e così via. Il POS tagging consiste nell'etichettare correttamente le parole in base al ruolo che la parola ricopre e ciò avviene in base a regole grammaticali e analisi di contesto.

Lemmatization è un'operazione che consiste nel riportare le forme flesse al lemma, ovvero la forma "base" di una parola.

Capitolo 5

Metodologia: una strategia basata sulle knowledge base

Come descritto nell'illustrazione delle varie strategie utilizzate negli articoli presentati nel capitolo 3, è possibile riassumere il task dividendolo in tre fasi: la fase di estrazione delle informazioni dal video, la fase di estrazione delle informazioni dal testo e l'associazione delle caratteristiche eterogenee.
Questi processi sono spesso molto onerosi sia da un punto di vista computazionale sia per ciò che concerne la quantità di memoria necessaria. La fase di training di una rete, infatti, prevede l'impiego di dataset di grandi dimensioni e tempi notevoli per l'apprendimento. Volendosi concentrare su un particolare aspetto della pipeline spesso si ricorre a reti pre-allenate ovvero modelli precedentemente addestrati su dataset vasti, che è possibile utilizzare sia senza effettuare modifiche sia riaddestrando solo gli ultimi layer della rete. Considerando per esempio una rete convoluzionale, i primi layer sono responsabili delle acquisizioni di elementi molto semplici e forme elementari. Prendendo una rete pre-allenata è possibile modificare l'apprendimento dei layer superiori per il riconoscimento di oggetti diversi da quelli previsti dal modello.

Nella strategia intrapresa, l'utilizzo delle reti pre-allenate costituisce un elemento fondamentale per l'estrazione delle informazioni dai media al fine di concentrarsi

sulle potenzialità dell'utilizzo di Wikidata. Per l'associazione di contenuti di natura differente mediante l'utilizzo delle entità ci si è ispirati a [38], articolo incentrato sull'associazione di audio e di testo di materiale educativo.

Per lo svolgimento del progetto è stato utilizzato Python come linguaggio di programmazione principale e SPARQL per interrogare Wikidata.

Python [1] è un linguaggio di programmazione di alto livello, ideato dall'informatico olandese Guido van Rossum e rilasciato nel 1991. Si tratta di un linguaggio orientato a oggetti intuitivo e di facile comprensione, rilasciato con licenza Open-Source. La sua diffusione è aumentata notevolmente nel tempo diventando, oggi, uno dei linguaggi di programmazione più utilizzati. Ulteriori vantaggi derivanti dalla scelta di Python sono costituiti dalla grande varietà di librerie e framework disponibili in particolare per ciò che riguarda l'ambito del machine learning.

SPARQL Simple Protocol and RDF Query Language, è un linguaggio utile a interrogare dati strutturati nel formato RDF.
RDF è stato standardizzato da W3C, World Wide Web Consortium, un'organizzazione internazionale che si occupa di migliorare l'accesso ai dati della rete e l'interazione con essi.
La ricerca di dati avviene attraverso delle query costituite da una tripletta soggetto-predicato-oggetto che indica lo schema e le relazioni caratteristiche dei dati da estrarre. La sintassi è simile a quella utilizzata da SQL e permette di vincolare il risultato utilizzando funzioni di analisi, termini di ordinamento e operatori booleani.

5.1 Estrazione delle informazioni dai video

Scegliendo di utilizzare pochi campioni radi per l'estrazione delle informazioni utili, da ogni video del dataset sono stati estratti frame a intervalli regolari pari a 3

[1] https://www.python.it/

secondi.

Ottenute le immagini si è proceduto all'analisi dei metodi per il riconoscimento degli oggetti principali presenti nelle scene. Inizialmente sono stati paragonati tre combinazioni di algoritmi e modelli per object detection pre-allenati con COCO.

Elenco delle classi di COCO				
person	bicycle	car	motorcycle	airplane
bus	train	truck	boat	traffic light
fire hydrant	stop sign	parking meter	bench	bird
cat	dog	horse	sheep	cow
elephant	bear	zebra	giraffe	backpack
umbrella	handbag	tie	suitcase	frisbee
skis	snowboard	sports ball	kite	baseball bat
baseball glove	skateboard	surfboard	tennis racket	bottle
wine glass	cup	fork	knife	spoon
bowl	banana	apple	sandwich	orange
broccoli	carrot	hot dog	pizza	donut
cake	chair	couch	potted plant	bed
dining table	toilet	tv	laptop	mouse
remote	keyboard	cell phone	microwave	oven
toaster	sink	refrigerator	book	clock
vase	scissors	teddy bear	hair drier	toothbrush

Tabella 5.1: Etichette messe a disposizione da COCO

COCO [39], acronimo di Common Objects in Context, è uno standard di riferimento per gli algoritmi di visione artificiale. COCO si pone come obiettivo il riconoscimento degli elementi posti all'interno di un contesto. L'individuazione degli oggetti in posizioni non ottimali, semi-occlusi o in ambienti complessi rende il task impegnativo. Le reti pre-allenate con COCO permettono il riconoscimento di 80 classi di oggetti di uso comune e di vario genere. Le etichette che possono essere riconosciute sono state elencate in tabella 5.1.

Ricercando una strategia di object detection che fosse veloce, ma anche accurata si sono effettuati tre tentativi di riconoscimento, due sfruttando ImageAI [40] e uno utilizzando darknet [31].

La libreria ImageAI [2] è uno strumento utile per sfruttare alcuni algoritmi di object detection. Tra questi sono stati individuati il modello RetinaNet e il modello YOLOv3 pre-allenati. Per l'utilizzo del modello è necessaria la creazione di un'istanza della classe "ObjectDetection" ed è possibile scegliere il modello da implementare, settandolo opportunamente.

```
detector.setModelTypeAsRetinaNet()
detector.setModelPath( os.path.join(execution_path , "
    resnet50_coco_best_v2.1.0.h5"))
```

```
detector.setModelTypeAsYOLOv3()
detector.setModelPath( os.path.join(execution_path , "yolo.h5"))
```

Parallelamente è stato implementato un ulteriore metodo che prevede l'utilizzo di openCV, libreria open-source per la computer vision. OpenCV mette a disposizione il metodo cv.dnn.readNet(...) che permette di leggere una rete di deep learning utilizzando il file di configurazione e il file dei pesi della rete permettendo di sfruttare quelli relativi a YOLOv3.

Per effettuare una prima analisi sono state scelte fotografie contenti alcuni degli oggetti corrispondenti alle label di COCO e mediante un'analisi diretta si è verificato la correttezza nell'individuazione dei bounding box e la capacità di rilevare correttamente il maggior numero di oggetti.

Tutte e tre le strategie sono accurate nel riconoscimento degli oggetti in ottica di estrazione di informazioni mirate all'identificazione del contenuto dei video, ma darknet si è dimostrata più efficiente in termini di tempistiche di elaborazione e ciò ha spinto a sceglierla per l'implementazione della metodologia presentata in questo elaborato. Un ulteriore vantaggio derivante da questa scelta è presentato dalla possibilità di sfruttare la versione di YOLOv3 pre-allenata con Open Images[3]

[2]https://imageai.readthedocs.io/en/latest/

[3]https://storage.googleapis.com/openimages/web/index.html

per ulteriori sperimentazioni. Open Images è un dataset creato da Google che comprende box per l'individuazione di 600 categorie alcune delle quali sono legate da relazioni gerarchiche.
In figura 5.1 è stato riportato un esempio di gerarchia tra le etichette appartenenti al ramo relativo alla label "person": tutti gli elementi foglia dell'albero sono a loro volta label corrispondenti a classi di Open Images.

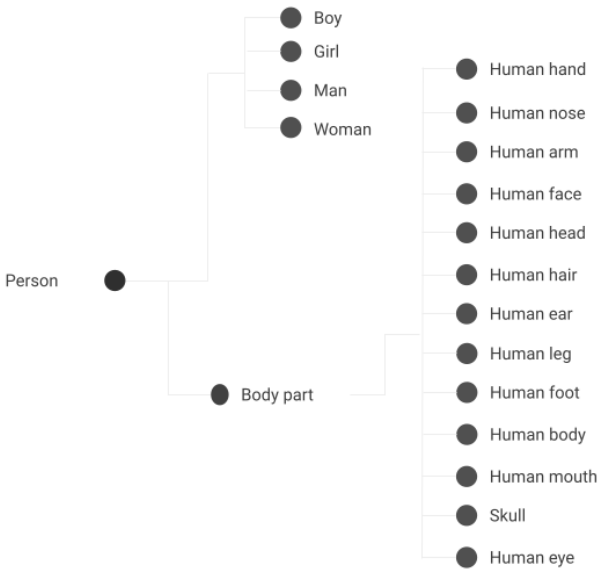

Figura 5.1: Esempio di gerarchia delle etichette di Open Images

I frame sono stati analizzati e per ognuno di quelli presi in esame sono stati individuati gli oggetti raffigurati tramite il sistema di object detection.

Per avere un ulteriore termine di paragone, una delle variazioni del metodo intrapreso prevede l'utilizzo dell'estrazione delle entità dai frame per mezzo delle Cloud Vision API di Google. Ci si aspetta che con questa strategia si possano riconoscere un numero maggiore di oggetti e con una maggior precisione.
L'API utilizzata permette l'estrazione di entità relative a Freebase, una base di conoscenza collaborativa a partire dalla quale è possibile passare alle entità di Wikidata attraverso la proprietà OWL:SAMEAS.

5.2 Estrazione delle informazioni dal testo

Analogamente a quanto accaduto in [38] per ogni caption sono state estratte le entità e ciò permette l'individuazione degli elementi principali che caratterizzano la descrizione dei video. Per estrarre le entità dalle caption sono stati utilizzati degli estrattori di entità quali TextRazor[4] e Babelfy[5] e le Cloud Natural Language API [6]. Dal momento che gli estrattori non derivano necessariamente le entità di Wikidata, queste sono state mappate nelle entità desiderate tramite la proprietà OWL:SAMEAS.

Poiché per un video possono essere presenti diverse caption, ci si è sincerati di rendere individuabili e riconoscibili i diversi testi tramite un opportuno identificativo.

5.3 Associazione dei contenuti

Per associare i contenuti derivanti dai video alle entità derivanti dal testo, si è proceduto creando un dizionario che associasse ad ogni label l'URL dell'entità di Wikidata corrispondente.

A partire dalle entità estratte dai video sono state cercate le entità figlie iterativamente attraverso la query:

```
SELECT ?parent ?child
WHERE{
    VALUES ?parent { <http://wikidata.org/entity/ID> }
    ?child wdt:P279|wdt:P361 ?parent.
}
```

[4]https://www.textrazor.com/

[5]http://babelfy.org/

[6]https://cloud.google.com/natural-language/docs/analyzing-entities

Nello specifico si è sfruttata la proprietà "subclass of" [7] caratterizzata dall'identificativo *P279* e la proprietà "part of" [8] indicata dal termine *P361* che permette di legare l'item all'oggetto di cui è parte.

La scelta di cercare le entità figlie deriva dall'osservazione di una maggiore generalità delle label di COCO rispetto alle entità estratte dal testo.

Da tutte le entità estratte dal testo sono stati derivati iterativamente i termini relativi alla proprietà "instance of" [9] secondo la query

```
SELECT ?instance ?class
WHERE{
    VALUES ?instance { <http://wikidata.org/entity/ID> }
    ?instance wdt:P31 ?class.
}
```

che permette di ricavare la classe di cui l'item è particolare esempio o membro.

Sia gli item derivanti dalle iterazioni effettuate a partire dalle etichette di COCO sia gli item derivanti dal testo sono stati salvati in modo da poterli leggere come dataframe, strutture tabulari della libreria *pandas* per l'organizzazione e la gestione dei dati caratterizzate da assi etichettati.

Successivamente si sono ricercate le associazioni potenziali delle entità derivate dal testo con tutte le possibili entità associate alle label di COCO.

Per ogni entità ricavata dal testo sono state effettuate diverse operazioni:

1. tentativo di associazione diretta con un'entità relativa alle label messe a disposizione da COCO

[7] https://www.wikidata.org/wiki/Property:P279

[8] https://www.wikidata.org/wiki/Property:P361

[9] https://www.wikidata.org/wiki/Property:P31

2. tentativo di associazione iterativa delle entità legate a quella presa in esame, tramite la proprietà "instance of", con una tra le 80 etichette ottenibili dall' operazione di object detection

3. ricerca dell'identificativo dell'entità di testo in analisi tra le entità figlie delle label di COCO, con l'obiettivo di risalire all'entità genitrice da cui deriva.

Avendo ottenuto tutte le possibili associazioni delle entità del testo con le entità derivanti dal video e avendo ricavato le entità indicative degli oggetti rilevati dai frame si è potuto valutare l'intersezione tra i set delle due modalità di contenuti. Da tale intersezione è possibile ottenere un valore indicativo della precisione dell'algoritmo utilizzato tramite un'opportuna metrica descritta in seguito.

La ricerca così effettuata ha il limite di considerare esclusivamente gli oggetti del video che sono presenti anche nella caption come entità, ma di non dare ulteriori informazioni utili quali ad esempio il contesto e l'azione.

Osservando i bounding box trovati nei frame si è notato che l'operazione di object detection dà più informazioni di quante non ne siano state sfruttate.
Per una migliore comprensione consideriamo la fotografia di una cucina. In questa fotografia sono stati evidenziati alcuni degli oggetti potenzialmente individuabili dal sistema di object detection.

Supponendo che l'immagine in figura 5.2 sia un frame di un video, difficilmente la descrizione associata farà riferimento ai singoli oggetti, come ad esempio ai vasetti o al forno o al tostapane a meno di uno scenario molto specifico come una pubblicità o di un'azione di interazione diretta con l'oggetto, mentre più probabilmente potrebbe essere utilizzato nella caption il termine "cucina" che però non verrebbe direttamente riconosciuto con il metodo utilizzato in precedenza. A partire da questa osservazione si è provato a verificare se una delle proprietà degli item potesse fornire informazioni di contesto e ciò a portato a individuare la proprietà *location* il cui identificativo è *P276*.

Figura 5.2: Immagine di esempio e potenziali bounding box

```
SELECT ?entity ?location
WHERE{
    VALUES ?entity { <http://wikidata.org/entity/ID> }
    ?entity wdt:P276 ?location.
}
```

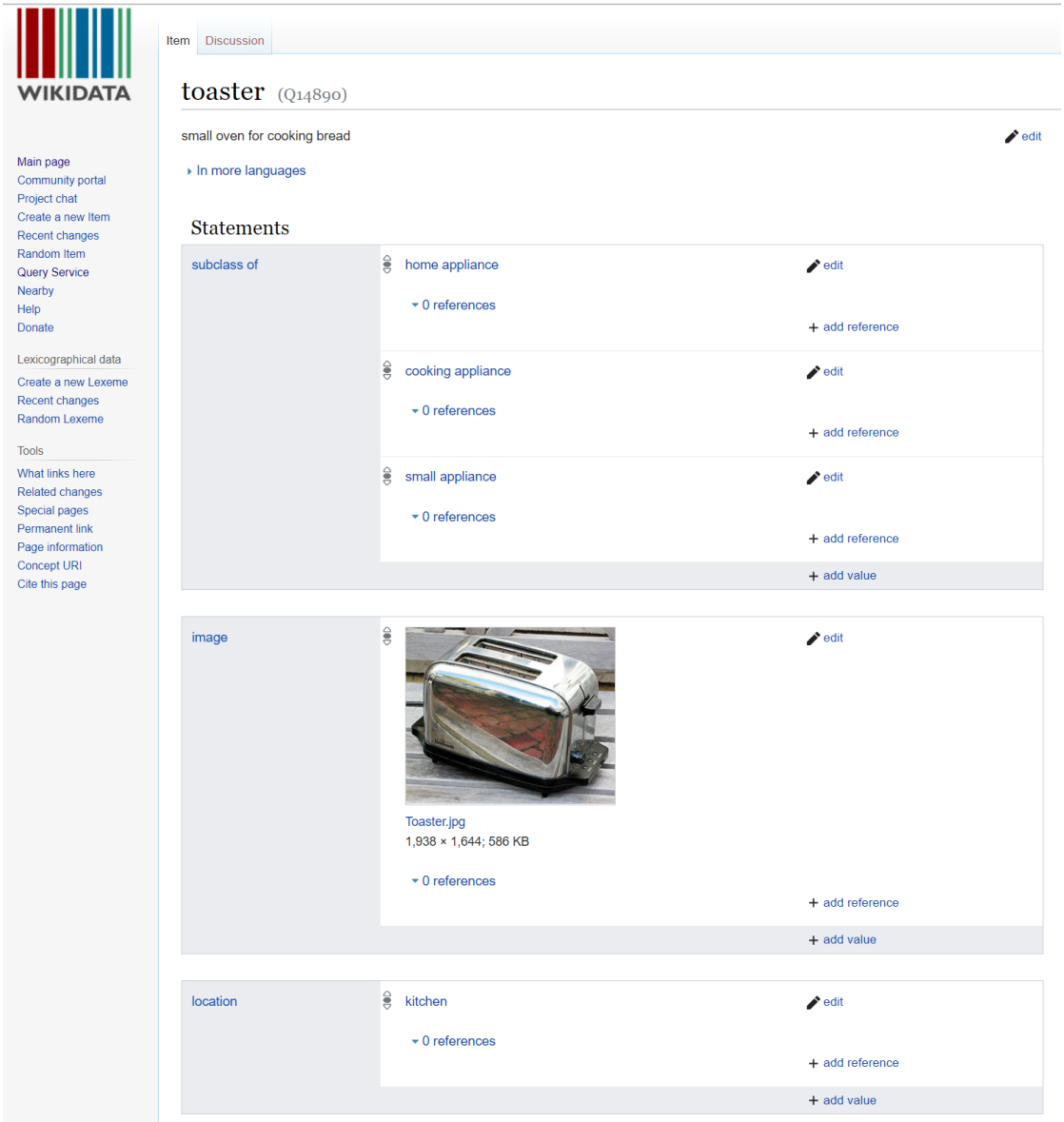

Figura 5.3: Screenshot della pagina di Wikidata dell'item "toaster"

Si consideri nuovamente l'esempio della fotografia della cucina e nello specifico l'item "toaster" (in figura 5.3 è stato riportato uno screenshot della pagina di Wikidata corrispondente). Effettuando un'opportuna query che leghi la location agli oggetti trovati è possibile aumentare la quantità di informazione restituendo l'entità relativa all'ambiente in cui generalmente gli oggetti si trovano.

L'informazione relativa alla location è stata estratta sia dalle entità del testo, sia dalle entità video e queste informazioni sono state aggiunte a quelle ottenute nel passaggio precedente. Per quanto riguarda il testo sono state prese in considerazione sia il prodotto derivante dall'applicazione della proprietà *location* agli item sia le entità del testo che a loro volta potevano essere termini indicanti una location individuabile tra i valori ottenuti a partire dal video.

Purtroppo pochi item posseggono la proprietà *location* e il potenziale non si palesa in modo evidente. Per esempio l'item "oven" non è completo di tale proprietà sebbene un tale oggetto possa essere indicativo del contesto in cui è inserito.

Il terzo tentativo di estensione dell'informazione deriva dall'idea di aumentare le probabilità di appaiamenti di video e testo corretti dando indicazioni sull'azione.

Sebbene esistano delle reti neurali in grado di analizzare le azioni rappresentate in un video, si è voluto approfondire la possibilità di derivarle a partire dalla knowledge base di Wikidata, in un processo meno oneroso da un punto di vista computazionale e di memoria.

Come accaduto in precedenza, la presenza di un oggetto di contesto riconosciuto dall'object detection, non determina la presenza dello stesso all'interno della caption, ma è possibile che vi sia un'azione legata allo stesso che venga espressa nella descrizione attraverso il verbo.
In figura 5.4 è stata riportata la tabella relativa alla proprietà *has use* [10] dell'item "oven". Tale proprietà lega l'oggetto agli usi principali che se ne possono fare ed è possibile sfruttare questa proprietà per legare i verbi delle caption alle label estratte dai frame.

[10]https://www.wikidata.org/wiki/Property:P366

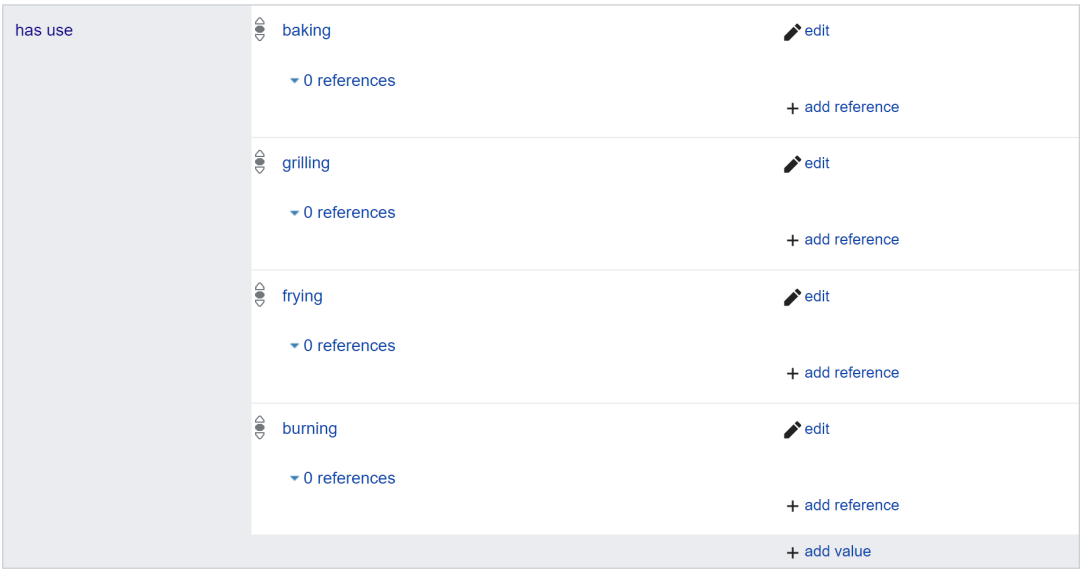

Figura 5.4: Screenshot della pagina di Wikidata della proprietà *has use* dell'item "oven"

Per derivare i verbi dalle caption si è sfruttata il toolkit NLTK [35] e il lettore di corpus WordNet [36] descritti nel capitolo precedente. In primo luogo è stato necessario analizzare le frasi ed estrarne i verbi. A tale scopo le frasi sono state divise in token e ogni token è stato etichettato in base alla parte del discorso corrispondente così da poter individuare solo i verbi le cui etichette hanno come lettera iniziale del tag "V". Per ogni verbo è stato individuato il lemma, ovvero la forma base, ed i verbi trovati sono stati associati all'identificativo della caption corrispondente.

Per tutte le entità associate alle label relative agli oggetti riconosciuti nei frame è stata ricercata la proprietà *has use* che indica i principali utilizzi dell'oggetto in esame.

Di seguito è stata riportata la query utile a derivare i valori della proprietà *has use*.

```
SELECT ?entity ?usage
WHERE{
    VALUES ?entity { <http://wikidata.org/entity/ID> }
    ?entity wdt:P366 ?usage.
}
```

La query riportata ritorna però l'URL identificativa del valore relativo alla proprietà in analisi da cui si sono estratte le label corrispondenti e per permettere l'associazione con i termini derivanti dal testo si è proceduto con la ricerca dei lemmi.

In questa fase ci si è sincerati di tenere traccia sia dell'item di provenienza che ha generato i valori per mezzo di quella proprietà sia degli identificativi dei valori stessi associati alle etichette, mediante la creazione di appositi dataframe. Ottenute le stringhe descrittive dei valori, queste sono state elaborate dal *lemmatizer* per ottenerne la forma base.
A questo punto si è verificato che fossero possibili associazioni tra le forme base ottenute dai verbi derivanti dalle caption con i lemmi ottenuti attraverso la proprietà *P366* e per mezzo un percorso a ritroso sui dataframe derivati dalle entità

dei video, si sono individuati gli identificati d'uso utili per il testo. Le indicazioni così ottenute sono state aggiunte alle precedenti per arricchire la rappresentazione dei contenuti nelle due modalità al fine di ottenere una descrizione più completa e permettere un'associazione delle azioni potenziali legate agli oggetti riconosciuti nei frame con i verbi descrittivi delle azioni presenti nelle caption.

Uno dei punti su cui ci si è interrogati riguardava l'esiguo numero di oggetti rilevabili dal sistema di object detection pre-allenato con COCO. L'ipotesi che a un numero maggiore di etichette ricavabili, corrispondesse una maggiore precisione associativa di video e testo ha indotto a tentare di estrarre informazioni visive utilizzando le label di Open Images che sono decisamente più numerose. Nel caso del primo esperimento, non vengono distinti per esempio gli adulti dai ragazzi e le donne dagli uomini e questa maggiore generalità induce a una minor discriminazione tra i contenuti: uno stesso video potrebbe essere associato a innumerevoli testi per il solo fatto di raffigurare una persona rendendo meno preciso l'accoppiamento, non avendo maggiori informazioni sull'individuo.

Utilizzando darknet, i file dei pesi e di configurazione sono stati impostati per l'utilizzo di Open Images e percorrendo gli stessi passi illustrati in precedenza, si è proceduto con l'estrazione delle entità e con i processi di arricchimento delle informazioni precedentemente descritti.

Già in fase di object detecton si è notato che sebbene fossero disponibili etichette specifiche per alcune categorie, come mostrato in figura 5.1, spesso venivano attribuite label più generiche e anche la quantità di oggetti riconosciuti non era maggiore rispetto a quelli rilevati precedentemente. Per confronto, nella tabella 5.2, sono stati riportati i valori identificativi per tre dei video analizzati (video 0, video 1 e video 10) paragonando i termini derivanti dal primo esperimento avente 80 label disponibili e il secondo avente 600 label disponibili.

Dalla tabella 5.2 è possibile notare come a un maggior numero di etichette non corrisponde necessariamente una maggior informazione. Rilevando meno oggetti anche la ricerca del contesto e dell'azione diventa più complessa e meno efficace e la discriminazione tra contenuti in fase di accoppiamento è resa meno significativa.

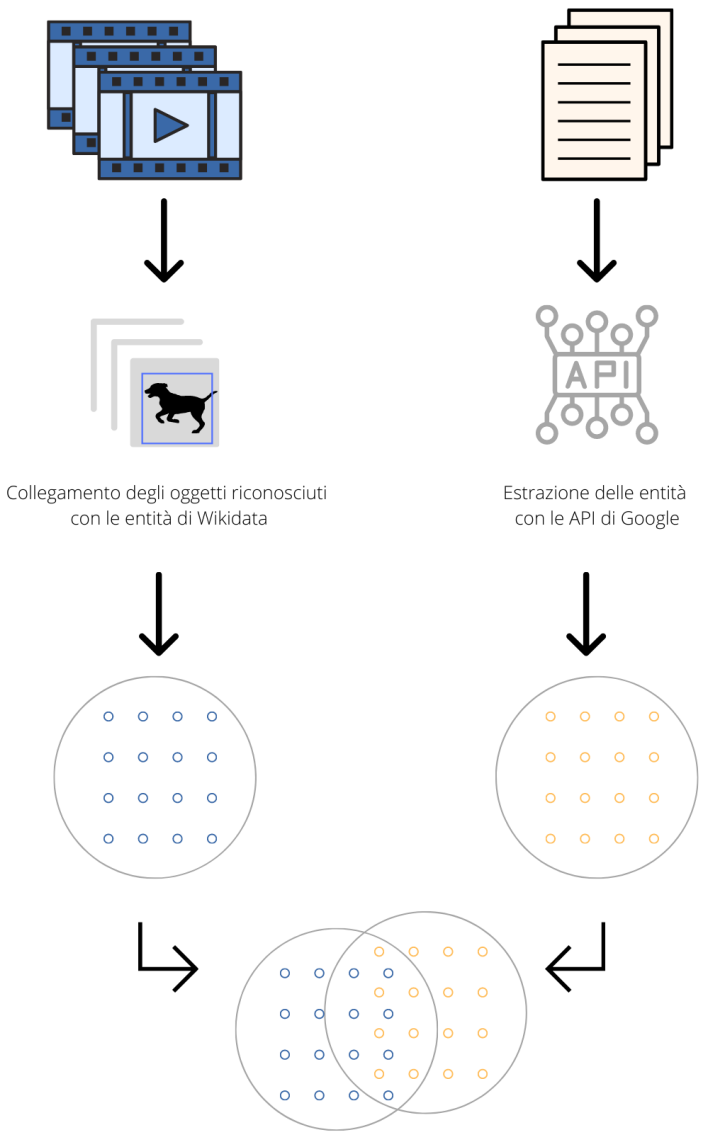

Figura 5.5: Schema riassuntivo della metodologia intrapresa

V 0 esp1	V 0 esp2	V 1 esp1	V 1 esp2	V 10 esp1	V 10 esp2
Q1420	Q756	Q81895	Q5	Q503	Q5
Q5	Q5	Q13276	-	Q5843	Q25416319
-	Q42889	Q80228	-	Q144	-
-	-	Q153988	-	Q3962	-
-	-	Q5	-	Q5	-
-	-	Q2425869	-	Q204776	-

Tabella 5.2: Tabella di confronto dei valori estratti dai frame con due diversi tipi di pre-training

Un' ulteriore variazione del metodo ha previsto l'utilizzo di un sistema che permetta di ottenere una più vasta gamma di entità ovvero le Cloud Vision API di Google per l' estrazione delle entità a partire dalle immagini. Questo sistema si suppone essere più preciso nel riconoscimento degli oggetti raffigurati nei frame permettendo di individuare molteplici elementi e rendendo più efficace l'associazione dei contenuti eterogenei.

Dal momento che non si è vincolati a un numero ristretto di etichette, ci si aspetta un maggior grado associazione e di precisione rispetto ai metodi utilizzati in precedenza.

L'aspetto negativo nell'utilizzo delle API di Google è rappresentato dall'impossibilità di effettuare un'analisi approfondita del funzionamento interno dal momento che questa tipologia di strumenti, a differenza di quelli open-source utilizzati in precedenza sono da considerarsi delle black box.

Per verificare che effettivamente l'estrazione delle entità e l'elaborazione eseguita con le modalità sopra descritte portino a un miglioramento della precisione dei risultati si è effettuato un ordinamento casuale dei documenti in fase di calcolo delle metriche per valutare i punteggi ottenuti.

Capitolo 6

Sezione sperimentale

6.1 Il dataset

L'analisi dello stato dell'arte ha permesso di individuare diversi dataset e MSR-VTT [41] risulta essere uno dei nomi che ricorre con più frequenza. Questo dataset è stato creato per risolvere il task di traduzione di contenuti video in contenuti testuali. Rispetto ad altri che collezionano video dai contenuti più specifici ad esempio YouCook, MSR-VTT comprende un'ampia collezione di video generici associati a molteplici descrizioni in linguaggio naturale. Le caption sono state definite dai dipendenti di Amazon Mechanical Turk che hanno fornito molteplici annotazioni per ogni video.

I video appartengono a 20 diverse categorie di vario contenuto tra le quali: sport, musica, moda, veicoli, film, cartoni animati, documentari, animali, show televisivi, pubblicità ed altro. Il numero totale dei video ammonta a 10000 mentre e le caption sono 142200

Il dataset è stato utilizzato separando i file video dai file di testo i quali sono stati inseriti in due pool differenti e analizzati in modo distinto in base al contenuto derivato. Solo in fase di calcolo delle metriche si è tenuto in considerazione il legame derivante dall'identificato ad essi associato.

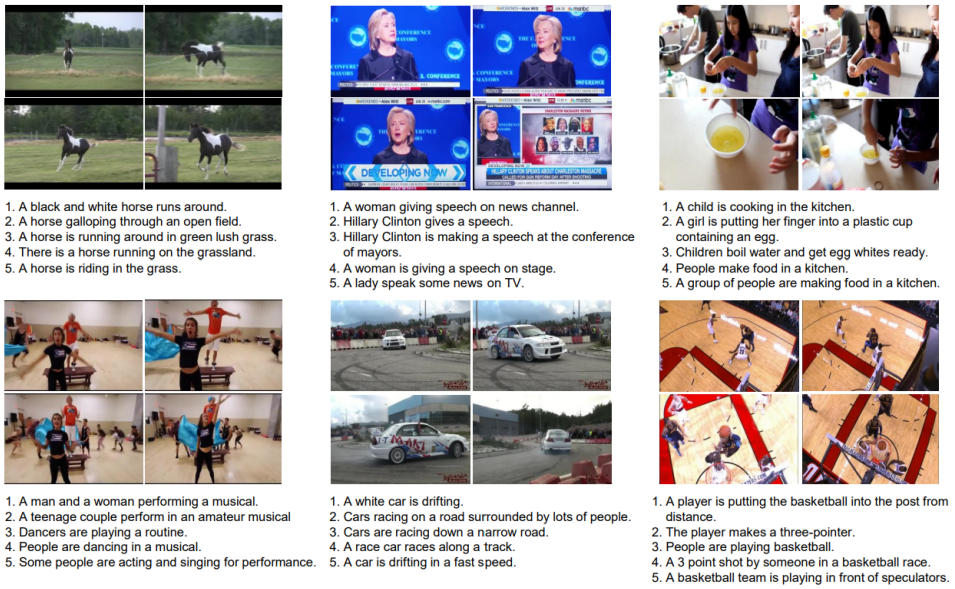

Figura 6.1: Esempio di video e testi associati di MSR-VTT [41]

6.2 Panoramica delle metriche

Per la valutazione delle sperimentazioni presentate in questo elaborato, è stata utilizzata una metrica chiamata mean average precision, mAP, utile a valutare la media dei punteggi di precisione derivanti da ogni query effettuata. Si tratta di un valore calcolato al fine di determinare la bontà di un sistema di information retrieval, ma per capirne il significato è necessario prima introdurre ulteriori definizioni [42].

Tra le metriche più conosciute vi sono la recall e la precision, introdotte da Cyril come metodi per misurare i sistemi di information retrieval [1].
Con il termine recall si intende il rapporto tra i documenti rilevanti recuperati correttamente per quella determinata query e l'intero insieme di documenti rilevanti, secondo la formula:

$$recall = \frac{|documenti\_rilevanti| \cap |documenti\_recuperati|}{|documenti\_rilevanti|}$$

Con il termine precision si intende invece il rapporto tra i documenti rilevanti recuperati correttamente per quella determinata query e i documenti recuperati:

$$precision = \frac{|documenti\_rilevanti| \cap |documenti\_recuperati|}{|documenti\_recuperati|}$$

Un elevato punteggio di recall indica, dunque, un grande quantitativo di documenti recuperati per una query, ma all'aumentare di questo valore potrebbe diminuire la precisione, dal momento che potrebbero essere considerati anche un maggior numero di documenti non rilevanti.

Nel caso in cui non si desideri considerare tutti gli elementi recuperati è possibile troncare ai primi k item più importanti: in questo caso si parla di recall@k o precision@k.

Per valutare i risultati restituiti dalla query tenendo conto dell'ordinamento in base al punteggio è utile utilizzare l'avarage precision AP:

$$AP = \frac{\sum_{k=1}^{n} precision@k \times rel@k}{numero\_documenti\_rilevanti}$$

dove rel@k è una funzione indicatrice che restituisce 1 in caso di pertinenza del documento recuperato e 0 altrimenti.

Volendo valutare il punteggio su più query contemporaneamente per avere un valore d'insieme è possibile utilizzare la mean avarage precision che media la somma dei punteggi di precisione calcolati per ogni query, q:

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

dove Q è il numero totale di query.

Al fine di determinare i punteggi sugli esperimenti effettuati è stato necessario utilizzare una libreria utile al calcolo del mean avarage precision. Questa libreria permette di paragonare liste di valori predetti e termini di ground truth per poi calcolarne il mAP@k.

Dal momento che si desidera considerare tutti i documenti a disposizione, si utilizza come valore di troncamento k la totalità dei video o dei testi disponibili a seconda della modalità con cui viene effettuata la query.

Poiché le liste mantengono l'informazione dell'ordine di inserimento dei valori è necessario effettuare un mescolamento randomico tra i documenti aventi lo stesso numero di termini di intersezione tra i set di entità di materiali eterogenei.

6.3 Riassunto delle variazioni dell'approccio intrapreso

Per una maggior comprensione delle fasi della metodologia presentata in questo elaborato, si intende riassumere le variazioni dell'approccio intrapreso.

In primo luogo si è provato ad effettuare un'associazione delle entità ottenute dai testi e dai video. Tale processo è stato reso possibile non solo per mezzo dell'associazione diretta tra gli identificativi delle entità di video e testo, ma anche mediante la ricerca delle entità figlie ottenute tramite le proprietà *subclass of* e *part of* di Wikidata a partire dalle label attribuite agli oggetti riconosciuti nei frame e mediante le classi derivate dalle entità estratte dal testo per mezzo della proprietà *instance of*.

Per un maggior arricchimento delle informazione si è proceduto ampliando il set di entità con le informazioni relative alla proprietà *location* attribuite ai vari item estratti in precedenza.

Nel tentativo di rendere maggiormente completa l'informazione dei contenuti dei media di diversa natura, si è presa in considerazione anche l'informazione relativa all'azione mediante la proprietà *has use* associata agli oggetti riconosciuti nei video e per mezzo dei verbi individuati nelle caption.

Mentre per quanto riguarda l'estrazione delle entità del testo è stato utilizzato esclusivamente il sistema di estrazione illustrato nel paragrafo 5.2, per tutte le variazioni del metodo, per ciò che concerne i video, il riconoscimento degli elementi è stato tentato tramite un sistema di object detection prima settato in modo tale da riconoscere istanze proprie delle 80 classi di COCO e in seguito in modo tale da riconoscere oggetti corrispondenti alle 600 classi di Open Images.

Per avere un ulteriore termine di paragone, in un secondo momento si è verificata la potenzialità della disponibilità di un maggior numero di label tramite l'impiego delle API di Google anche per l'estrazione delle entità dai frame. In questo caso l'associazione è avvenuta solo per confronto diretto delle entità estratte da video e testo.

Infine, per verificare l'efficacia del sistema e per accertarsi che la metodologia intrapresa apportasse dei benefici al task si è effettuato un calcolo della precisione in seguito all'ordinamento casuale degli elementi del dataset in fase di calcolo delle metriche.

Ogni variazione della metodologia è stata valutata sia per il task di video-to-text retrieval sia per task di text-to-video retrieval.

Poiché l'estrazione delle entità dal testo è stata effettuata con un'unica modalità, nella tabella 6.1 sono stati presi in considerazione i diversi metodi di estrazione delle informazioni dai frame.

Darkent(COCO)	Darknet(Open Images)	Vision API	Random
Entità	Entità	Entità	Elem dataset
Entità+location	Entità+location	-	-
Entità+loc+azione	Entità+loc+azione	-	-

Tabella 6.1: Tabella di confronto delle variazioni del metodo in base alla modalità di estrazione e associazione delle informazioni dai video

6.4 Risultati

L'obiettivo perseguito in questo elaborato riguardante il video-to-text retrieval e il text-to-video retrieval consiste nell'associare correttamente i video alle caption corrispondenti scelte all'interno di un pool e viceversa. Avendo effettuato diverse sperimentazioni a tal fine si intende verificare l'efficacia dei metodi paragonando i valori ottenuti dal calcolo del mAP.

Basandosi sull'aspettativa per la quale l'insieme intersezione del set di entità ottenute dal video o dal testo di query sia massima quando confrontato con il set relativo alla caption o al video ad esso associato, si procede con il calcolo dei punteggi.

Esperimenti	video-to-text	text-to-video
COCO-Ent	0,002924	**0,023267**
COCO-Ent+loc	0,002940	0,022630
COCO-Ent+loc+azione	0,002754	0,021191
Open Images-Ent	0,001289	0,019479
Open Images-Ent+loc	0,001289	0,019479
Open Images-Ent+loc+azione	0,001257	0,018202
Vision API	**0,010329**	0,019407
Random	0,000492	0,001389

Tabella 6.2: Score relativi a tutte le variazioni della strategia utilizzata ottenuti calcolando il mean avarage precision

Dalla tabella 6.2 è possibile notare che i valori più alti sono stati ottenuti in fase di video-to-text retrieval in corrispondenza dell'impiego delle API di Google per l'etichettatura degli elementi presenti nei frame, mentre il punteggio maggiore per il task di text-to-video retrieval si è ottenuto per mezzo dell'associazione dei contenuti visivi e testuali utilizzando le entità relative alle label di COCO.

La ricerca dei video a partire da query testuali beneficia maggiormente dell'approccio intrapreso dal momento che i valori di mAP risultano maggiori rispetto ai corrispettivi ottenuti a partire da query in forma di video.

Seppur facendo registrare oscillazioni più o meno significative, l'utilizzo delle entità ha permesso di raggiungere punteggi maggiori rispetto a quelli ottenuti mediante l'ordinamento casuale dei risultati.

6.4.1 Discussione dei risultati ottenuti

Ciò che risulta evidente dalla tabella 6.2 è che i valori relativi al recupero del testo a partire dal video sono decisamente più bassi rispetto ai corrispettivi valori di mAP ottenuti utilizzando le caption come query. Considerando che a ogni video corrispondono più caption, ma che a un testo corrisponde un solo video, la giustificazione a tale discrepanza può essere individuata nella definizione della metrica.

Nel caso del video-to-text-retrieval, quando viene definita la funzione AP_k =

metrics.apk(...) il confronto tra le liste effettive e teoriche restituirà un punteggio che terrà conto di tutti i contributi derivanti da ogni caption associata al video, in base all'ordinamento globale dei documenti. Se anche solo una minima parte dei testi non ha tra i set di entità valori di intersezione sufficientemente elevati, essi verranno spinti in fondo alla lista effettiva totale abbassando notevolmente il punteggio della metrica. Perché il mAP sia massimo, tutti e solo i testi coerenti devono restituire un set di intersezione massimo. Si consideri inoltre la definizione della metrica AP. Tale espressione presenta a denominatore il numero dei documenti rilevanti e poiché a un video corrispondono diversi testi, mentre a un testo corrisponde un solo video, vi è una discrepanza che influisce sul punteggio finale.

Si considerino ora i valori delle varie fasi dell'esperimento effetuato per mezzo delle label di COCO, sebbene gli score non presentino discrepanze così accentuate tra i tre risultati, si notano piccole oscillazioni dei valori.
Per quanto riguarda il video-to-text retrieval il punteggio massimo si ottiene integrando le informazioni di location alle precedenti, mentre il punteggio minore è restituito valutando anche le informazioni di azione. Anche nel caso del text-to-video retrieval si nota una diminuzione degli score in corrispondenza dell'aumento delle informazioni, indice della creazione di set più generici che rendono associabile una query a più documenti.

Per entrare nel dettaglio della problematica occorre considerare che le indicazioni di uso derivanti dalla proprietà *has use* di Wikidata non restituiscono necessariamente l'intero range di utilizzi possibili dell'item in esame. Il problema si presenta quando un'azione può essere associata ad oggetti diversi, ma non tutti sono provvisti del valore di interesse in corrispondenza della proprietà *has use*, oppure nel caso in cui manchino totalmente dell'informazione di utilizzo. Si pensi per esempio all'item "bed", tra i valori relativi alla proprietà *has use* è stato inserito il valore "sitting" il quale però si addice anche al termine "chair", "bench" oppure "couch", sebbene questi non siano provvisti della tabella relativa al termine *has use*. Un'ipotetica caption contente come verbo "sitting" non verrà legata per mezzo dell'azione all'immagine in cui è stata individuata una sedia o una panchina, al contrario la descrizione del video verrà collegata a un'immagine contente un letto.

A tale disparità e asimmetria di informazioni consegue l'introduzione di rumore durante il tentativo di integrazione dei contenuti per mezzo dell'azione e ciò porta a un peggioramento della precisione nel task di retrieval.

Per questo esperimento di integrazione, ad ogni modo, i valori non si discostano al punto da dare indicazioni più precise.

Le sperimentazioni effettuate utilizzando una rete pre-allenta con Open Images, in grado di riconoscere un maggior numero di classi, non hanno portato alla più precisa descrizione dei contenuti. L'utilizzo di un maggior numero di label avrebbe dovuto permettere una maggiore discriminazione tra gli oggetti migliorando la precisione nell'associazione dei media eterogenei.

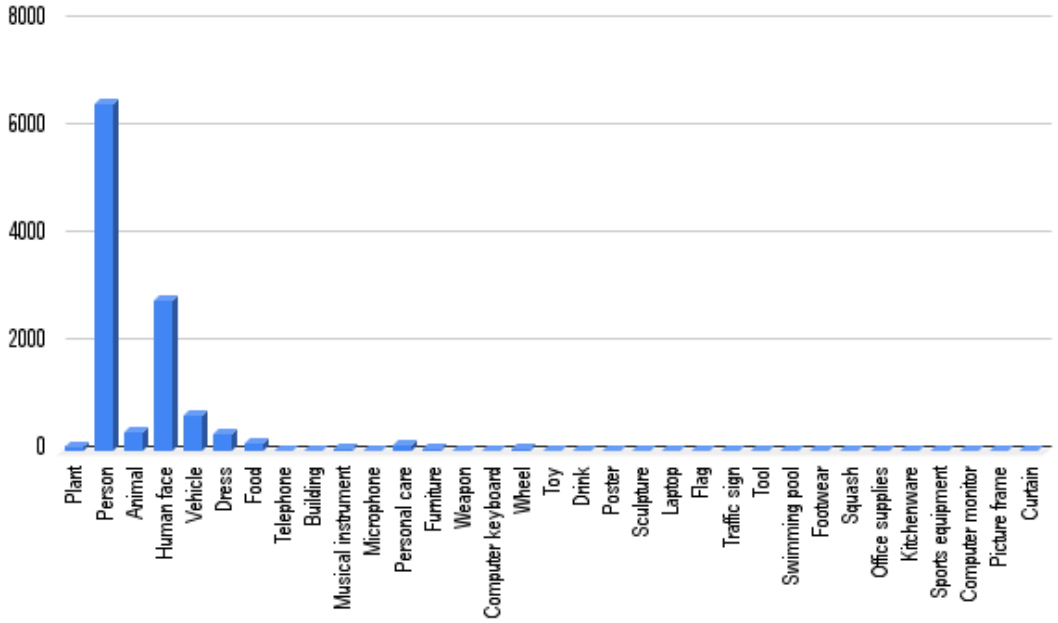

Figura 6.2: Istogramma delle etichette ottenute dall'analisi con Open Images

In figura 6.2 è stato riportato un istogramma avente sulle ascisse le etichette estratte dai frame e sulle ordinate il valore totale delle istanze trovate. Graficamente è immediato notare il numero esiguo di classi di oggetti riconosciute rispetto alle 600 categorie disponibili. I termini più specifici lasciano il posto ad etichette più

generiche rendendo la variabilità degli oggetti inefficace in ottica di discriminazione dei contenuti.

Il numero di etichette individuate dal primo sistema è nettamente maggiore, tanto che sono state trovate istanze facenti parte di 79 classi su 80 disponibili per la rete pre-allenata con COCO.

In tabella 6.3 sono state riportate le classi individuate nella prima esperienza e il relativo numero di occorrenze.

Paragonando i risultati ottenuti riportati nel grafico con la variabilità delle categorie riportate in tabella 6.3, è evidente che il primo esperimento, effetuato mediante le etichette di COCO, ha permesso di ottenere istanze più specifiche dal momento che, per esempio, sono stati distinti i diversi tipi di animali, mentre nella seconda esperienza tutti gli animali individuati sono stati raggruppati sotto l'etichetta "animale".

I set di entità dell'esperimento che ha visto l'impiego delle etichette di Open Images sono stati ottenuti con un procedimento analogo al precedente e paragonando i risultati dell'esperimento in tabella 6.2 è interessante notare come il contributo dell'informazione relativa alla location sia del tutto irrilevante nel caso in analisi. Indipendentemente dalle etichette utilizzate in fase di object detection, l'aggiunta dell'informazione dell'azione ha abbassato i punteggi di mAP.

Anche in questo caso si evidenzia una discrepanza dei punteggi ottenuti nel caso del video-to-text retrieval e nel caso del text-to-video retrieval. Coerentemente con quanto riportato nell'analisi dei risultati dell'esperimento condotto per mezzo di YOLOv3 pre-allenato con COCO, anche in questo caso gli score sono maggiori nel caso del recupero dei video a partire da una query testuale, mantenendo una differenza dell'ordine di grandezza tra i due task pari a un fattore 10.

Per avere un quadro più completo del funzionamento nel campo del video-to-text retrieval e del text-to-video retrieval per mezzo delle entità, si è proceduto verificando il grado di associabilità delle entità ottenute per mezzo dei servizi di Google utilizzati in fase di estrazione delle informazioni dal video con le entità derivate dalle caption. Come spiegato in precedenza a una maggiore varianza delle etichette dovrebbe corrispondere una maggior precisione nell'accoppiamento degli

Label	Num.	Label	Num	Label	Num
Car	793	Spoon	155	Banana	49
Dog	359	Person	8059	Baseball bat	40
Tie	868	Cup	570	Truck	30
Tv	650	Apple	73	Book	223
Cake	219	Bowl	536	Refrigerator	129
Bird	145	Potted plant	187	Backpack	108
Suitcase	48	Donut	32	Cell phone	352
Laptop	275	Surfboard	75	Keyboard	35
Boat	71	Sports ball	165	Clock	136
Horse	175	Handbag	95	Tennis racket	43
Cat	113	Orange	37	Bicycle	124
Elephant	18	Fork	40	Bus	87
Bottle	360	Remote	60	Fire hydrant	22
Toilet	17	Bear	40	Traffic light	75
Oven	155	Wine glass	91	Knife	88
Scissors	59	Sheep	15	Train	74
Motorcycle	159	Bed	65	Sink	65
Giraffe	6	Skateboard	44	Carrot	45
Parking meter	22	Sandwich	20	Dining table	340
Umbrella	108	Toothbrush	47	Airplane	29
Skis	6	Teddy bear	23	Stop sign	24
Microwave	83	Mouse	17	Vase	109
Chair	704	Broccoli	20	Frisbee	83
Kite	4	Snowboard	11	Cow	50
Bench	89	Pizza	54	Zebra	9
Hot dog	12	Baseball glove	56	Toaster	1
Couch	104	-	-	-	-

Tabella 6.3: Oggetti individuati dal sistema di object detection e numero di occorrenze

item, così come una più precisa capacità di riconoscimento degli oggetti permette di ridurre l'errore nella descrizione del contenuto dei documenti.

I punteggi ottenuti permettono di effettuare alcune osservazioni di importanza notevole. In primo luogo, sebbene vi sia una differenza tra il valore di mAP nel caso del video-to-text retrieval e del text-to-video retrieval questa risulta decisamente inferiore rispetto a quella che si osserva nei casi precedenti. In secondo

luogo, mentre i valori relativi alle query in forma visuale sono superiori a qualsiasi esperimento precedente, i punteggi ottenuti a partire da query in forma testuale risultano paragonabili o addirittura inferiori rispetto alle variazioni precedenti.

I valori ottenuti mediante l'ordimento casuale della lista effettiva in fase di calcolo delle metriche, ha restituito punteggi inferiori di un fattore 10 rispetto alle precedenti, indice del fatto che il lavoro portato avanti con la metodologia esposta in questo elaborato contribuisce positivamente al task. È interessante notare come anche in questo caso vi sia una differenza di valori tra le due tipologie di retrieval a conferma dell'ipotesi avanzata precedentemente del fatto che la causa della differenza di punteggi tra task di video-to-text e text-to-video retrieval sia da ricercare nella differenza del numero di item da recuperare per una singola query.

I punteggi relativi agli esperimenti permettono di ragionare su eventuali sviluppi futuri della metodologia intrapresa in ottica di miglioramento delle prestazioni del sistema.

Capitolo 7

Conclusioni e sviluppi futuri

7.1 Conclusioni

Nel presente elaborato si è effettuata una panoramica degli strumenti maggiormente utilizzati per il raggiungimento dei task di video-to-text retrieval e text-to-video retrieval. La sfida dell'associazione di materiali eterogenei al fine di recuperare da un insieme solo i media realmente coerenti a partire da una query di diversa natura interessa il campo della ricerca e numerose sono le soluzioni efficaci.

La metodologia proposta ha incluso l'utilizzo di reti pre-allenate, servizi di Google Cloud, framework e altri strumenti, ma il punto focale dell'approccio è rappresentato dall'utilizzo di una knowledge base per collegare elementi altrimenti difficilmente associabili direttamente.

Per la fase di estrazione delle entità dai video, sono stati testati differenti metodi per il riconoscimento degli oggetti all'interno dei frame.
L'ipotesi per cui a un numero maggiore di classi riconoscibili consegue una maggior precisione associativa dei media eterogenei è stata testata sia mediante l'utilizzo di reti pre-allenate per l'object detection che permettessero il riconoscimento di un numero diverso di classi, (80 messe a disposizione da COCO e 600 messe a disposizione da Open Images), sia per mezzo delle Cloud Vision API di Google. Mentre per i sistemi di object detection la mappatura delle entità è avvenuta in

modo manuale, l'associazione degli oggetti alle entità di Wikidata ottenute per mezzo dei servizi di Google è stata ottenuta in modo automatico.

In corrispondenza dell'impiego dei sistemi di object detection pre-allenati con COCO e Open Images si è esplorata la possibilità che l'aggiunta dell'informazione di contesto ottenuta per mezzo della proprità *location* di Wikidata apportasse benefici in termini di precisione associativa. In un secondo momento è stata aggiunta anche l'informazione relativa all'azione ricavata dalle label relative agli oggetti riconosciuti nei frame per mezzo della proprietà *has use* di Wikidata.

L'estrazione delle entità testuali è avvenuta grazie all'impiego di strumenti quali TextRazor[1], Babelfy[2] e Cloud Natural Language API di Google[3]. In fase di arricchimento delle informazioni si è sfruttata la proprietà *location* per la derivazione del contesto ed è stato impiegato il toolkit NLKT per l'estrazione dei verbi dalle caption per la deduzione dell'azione.

Gli score di mAP ricavati hanno evidenziato una discrepanza di valori tra i task di video-to-text e text-to-video retrieval, dal momento che i primi risultano inferiori ai secondi nella totalità dei casi. Nel caso di query in forma di video, il punteggio maggiore si è registrato in occasione dell'impiego delle Vision API di Google, mentre nel caso di query in forma testuale i risultati maggiori si sono ottenuti nell'esperimento che ha visto l'impiego del sistema di object detection pre-allenato con COCO, sebbene l'aggiunta delle informazioni di contesto e azione abbiamo fatto diminuire i punteggi.

Il fatto che, almeno per il task di text-to-video retrieval, i punteggi ottenuti per mezzo di reti pre-allenate abbiano permesso il raggiungimento di score paragonabili o superiori a quelli ottenuti mediante l'utilizzo esclusivo dei servizi di Google permette di pensare ad un'eventuale sostituzione di questi, i quali sono da considerasi come

[1]https://www.textrazor.com/

[2]http://babelfy.org/

[3]https://cloud.google.com/natural-language/docs/analyzing-entities

black box, in favore di metodi esplorabili nel loro funzionamento. A partire da questa osservazione è possibile pensare a diversi sviluppi futuri attuabili mediante modifiche della pipeline di estrazione delle entità dai media soppiantando i servizi di Google Cloud con modelli open-source. Inoltre, è possibile migliorare il metodo anche mediante modifiche nella fase di associazione delle entità aggiungendo pesi diversi alle informazioni ottenute o integrando elementi mediali non ancora considerati, quali ad esempio l'audio.

7.2 Sviluppi futuri

7.2.1 Modifiche della pipeline di analisi testuale

Durante l'analisi delle risorse utili alla preparazione degli esperimenti si è cercato di utilizzare quanto più possibile strumenti open-source, almeno per ciò che concerne l'analisi dei contenuti visivi. Un primo tentativo di modifica del progetto consiste nella sostituzione della pipeline di estrazione delle entità del testo con modelli open-source pre-allenati, sostituendo le componenti dei servizi di Google con modelli esaminabili nei loro meccanismi di funzionamento. Il passaggio a sistemi aperti dà la possibilità di sperimentare maggiormente in termini di modifica dell'architettura o tipologia di allenamento nonché di comprendere più a fondo i meccanismi di funzionamento delle reti.

7.2.2 Modifiche della pipeline di analisi dei video

Per quanto riguarda la pipeline di analisi dei video gli sviluppi possono essere molteplici. Come si è evidenziato nei capitoli precedenti il numero degli elementi individuabili dal sistema di object detection, nonché la specificità degli stessi contribuiscono a discriminare maggiormente i contenuti rendendo l'associazione di testo e video più precisa. Se, per esempio, il sistema è in grado di riconoscere solo gli esseri umani, tutti i documenti contenenti una qualsiasi tipologia di persona saranno coinvolti, mentre potendo distinguere il genere o la fascia di età della stessa, sia in corrispondenza del testo sia relativamente al video, la discriminazione tra le risorse sarà maggiore e di conseguenza la precisione nel recupero dei media.

Come suggerito dalla presentazione dello stato dell'arte per un'analisi più completa dei contenuti è possibile utilizzare CLIP, abbandonando l'uso di un numero ristretto di label in favore di un apprendimento diretto del contenuto a partire dal testo correlato.

Una maggiore quantità di label disponibili non conoscibili a priori rende necessario automatizzare l'attribuzione degli identificativi di Wikidata. A partire dalle label è possibile effettuare una ricerca delle entità corrispondenti, ma comporta il rischio di associare anche URL non propriamente consone aumentando la quantità di rumore dell'informazione e rendendo la capacità di associazione di media eterogenei più complessa e meno precisa. Si rende necessario affinare il processo di disambiguation per risolvere i conflitti tra i termini individuati dal sistema.

7.2.3 Riconoscimento dell'azione

Dagli score ottenuti si è evidenziato un contributo non decisivo dell'aggiunta dei termini derivanti dai verbi delle caption e dalla proprietà *has use* utilizzata per estrarre l'uso principale degli oggetti. Attualmente gli item non sono completi di tutte le informazioni e come discusso in precedenza questo rende i risultati meno precisi. Wikidata è però una knowledge base modificabile da qualunque utente ed è auspicabile che una sua crescita permetta di ottenere informazioni sempre più complete.

Per ottenere un diverso termine di paragone per ciò che concerne la descrizione delle azioni è possibile provare a utilizzare delle reti neurali per l'action recognition e sfruttare Wikidata per ampliare l'informazione relativa. La creazione di un sistema statistico per l'associazione delle entità derivate dagli oggetti riconosciuti e delle azioni individuate potrebbero essere utili per l'ordinamento globale dei contenuti rendendo l'associazione dipendente non solo dalla grandezza dell'intersezione dei set, ma anche dai pesi individuati in base al legame tra entità e azione.

7.2.4 Integrazione di ulteriori media

In questo elaborato ci si è concentrati esclusivamente su due tipologie di media e nello specifico sui contenuti testuali e sui contenuti video, mentre non si è

considerata la componente audio. Per una maggiore completezza delle informazioni una strada percorribile è rappresentata dalla codifica di questa componente. Il contributo che la componente audio può dare dipende molto dalla tipologia del dataset utilizzato. Se si considerano per esempio dei dataset formati da videolezioni come accaduto in [38], il parlato è parte fondamentale del contenuto, e porta con sé una grande quantità di informazioni. Anche tipologie di video come ad esempio le pubblicità possono contenere termini chiave per il riconoscimento delle entità direttamente collegabili al testo o al video, mentre in caso di musica o suoni di contesto tale associazione può essere più complessa e le knowledge base possono costituire un elemento chiave per la riconducibilità dei suoni alle scene presentate.

7.2.5 Impiego della somiglianza semantica per il calcolo dei punteggi

Dall'analisi dei risultati si è evidenziata una discrepanza significativa tra i task di retrieval aventi come query contenuti testuali e task aventi come query contenuti video.

Poiché a un video possono essere associate diverse caption e che i testi possono essere simili tra loro se descrittivi di uno stesso video, è possibile considerare la somiglianza semantica delle caption al fine di regolare il peso dei testi che condividono contenuti semanticamente legati.

Da [43] possono essere colti alcuni spunti sul modo di procedere nel considerare le somiglianze semantiche. Uno studio interessante potrebbe concernere come varino i punteggi di precisione pesando opportunamente testi dai contenuti paragonabili. Idealmente verrebbero avvicinati nella lista dei documenti recuperati quei testi che condividono il contenuto, ma in caso di video non nettamente dissimili, anche le rispettive caption verrebbero pesate maggiormente, salendo nella lista dei testi ritenuti corretti con maggior probabilità. Bisogna però considerare che è più probabile trovare tutte le caption corrette all'interno di un insieme di contenuti simili piuttosto che analizzando esclusivamente la grandezza del set di intersezione.

Bibliografia

[1] Michael E. Lesk. «THE SEVEN AGES OF INFORMATION RETRIEVAL». In: 1998 (cit. alle pp. 1, 59).

[2] Vannevar Bush. «As We May Think». In: *Atlantic Monthly* 176.1 (mar. 1945), pp. 641–649. ISSN: 1072-5520. DOI: 10.1145/227181.227186. URL: http://www.theatlantic.com/doc/194507/bush (cit. a p. 1).

[3] Yuxin Peng, Xin Huang e Yunzhen Zhao. *An Overview of Cross-media Retrieval: Concepts, Methodologies, Benchmarks and Challenges*. 2017. arXiv: 1704.02223 [cs.MM] (cit. alle pp. 2, 5).

[4] David E. Rumelhart, Geoffrey E. Hinton e Ronald J. Williams. «Learning representations by back-propagating errors». In: *Nature* 323 (1986), pp. 533–536 (cit. a p. 8).

[5] Michael Phi. *Illustrated Guide to Recurrent Neural Networks*. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9. 2018 (cit. a p. 8).

[6] Michael Phi. *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21. 2018 (cit. alle pp. 8–10).

[7] Sepp Hochreiter e Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. a p. 9).

[8] Jianfeng Dong, Zhongzi Long, Xiaofeng Mao, Changting Lin, Yuan He e Shouling Ji. «Multi-level Alignment Network for Domain Adaptive Cross-modal Retrieval». In: *Neurocomputing* 440 (2021), pp. 207–219. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.01.114. URL: https://www.sciencedirect.com/science/article/pii/S0925231221002150 (cit. alle pp. 9, 13, 15, 25).

[9] A. Graves e J. Schmidhuber. «Framewise phoneme classification with bidirectional LSTM networks». In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* Vol. 4. 2005, 2047–2052 vol. 4. DOI: 10.1109/IJCNN.2005.1556215 (cit. a p. 10).

[10] Savelie Cornegruta, Robert Bakewell, Samuel Withey e Giovanni Montana. *Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks*. 2016. arXiv: 1609.08409 [cs.CL] (cit. a p. 10).

[11] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk e Yoshua Bengio. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. DOI: 10.48550/ARXIV.1406.1078. URL: https://arxiv.org/abs/1406.1078 (cit. a p. 10).

[12] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang e Meng Wang. «Dual Encoding for Video Retrieval by Text». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), pp. 1–1. DOI: 10.1109/TPAMI.2021.3059295 (cit. alle pp. 10, 13, 24).

[13] Tomas Mikolov, Kai Chen, Greg Corrado e Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL] (cit. alle pp. 11, 13).

[14] Rikiya Yamashita, Mizuho Nishio, Richard K. G. Do e Kaori Togashi. «Convolutional neural networks: an overview and application in radiology». In: *Insights into Imaging* 9 (2018), pp. 611–629 (cit. alle pp. 12, 14).

[15] Alex Krizhevsky, Ilya Sutskever e Geoffrey E Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». In: *Advances in Neural Information Processing Systems*. A cura di F. Pereira, C. J. C. Burges, L. Bottou e K. Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. a p. 12).

[16] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. 2018 (cit. a p. 12).

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville e Yoshua Bengio. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML] (cit. a p. 14).

[18] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta e Anil A. Bharath. «Generative Adversarial Networks: An Overview». In: *IEEE Signal Processing Magazine* 35.1 (gen. 2018), pp. 53–65. ISSN: 1053-5888. DOI: 10.1109/msp.2017.2765202. URL: http://dx.doi.org/10.1109/MSP.2017.2765202 (cit. a p. 14).

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser e Illia Polosukhin. «Attention is All you Need». In: *Advances in Neural Information Processing Systems*. A cura di I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan e R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf (cit. alle pp. 15, 16).

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee e Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL] (cit. a p. 17).

[21] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV] (cit. a p. 18).

[22] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV] (cit. alle pp. 19, 20).

[23] Han Fang, Pengfei Xiong, Luhui Xu e Yu Chen. *CLIP2Video: Mastering Video-Text Retrieval via Image CLIP*. 2021. arXiv: 2106.11097 [cs.CV] (cit. alle pp. 20, 21).

[24] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang e Dong Shen. *Improving Video-Text Retrieval by Multi-Stream Corpus Alignment and Dual Softmax Loss*. 2021. arXiv: 2109.04290 [cs.CV] (cit. alle pp. 21, 22).

[25] Zijian Gao, Jingyu Liu, Sheng Chen, Dedan Chang, Hao Zhang e Jinwei Yuan. *CLIP2TV: An Empirical Study on Transformer-based Methods for Video-Text Retrieval*. 2021. arXiv: 2111.05610 [cs.CV] (cit. alle pp. 23, 24).

[26] R. Hadsell, S. Chopra e Y. LeCun. «Dimensionality Reduction by Learning an Invariant Mapping». In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. 2006, pp. 1735–1742. DOI: 10.1109/CVPR.2006.100 (cit. a p. 24).

[27] Jianwei Yang, Yonatan Bisk e Jianfeng Gao. *TACo: Token-aware Cascade Contrastive Learning for Video-Text Alignment*. 2021. arXiv: 2108.09980 [cs.CV] (cit. a p. 26).

[28] Yali Amit, Pedro Felzenszwalb e Ross Girshick. «Object Detection». In: *Computer Vision: A Reference Guide*. Cham: Springer International Publishing, 2020, pp. 1–9. ISBN: 978-3-030-03243-2. DOI: 10.1007/978-3-030-03243-2_660-1. URL: https://doi.org/10.1007/978-3-030-03243-2_660-1 (cit. a p. 27).

[29] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He e Piotr Dollár. «Focal Loss for Dense Object Detection». In: *CoRR* abs/1708.02002 (2017). arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002 (cit. a p. 30).

[30] Joseph Redmon, Santosh Divvala, Ross Girshick e Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV] (cit. a p. 31).

[31] Joseph Redmon. *Darknet: Open Source Neural Networks in C*. `http://pjreddie.com/darknet/`. 2013–2016 (cit. alle pp. 32, 43).

[32] Joseph Redmon e Ali Farhadi. *YOLOv3: An Incremental Improvement*. 2018. arXiv: `1804.02767 [cs.CV]` (cit. a p. 32).

[33] Joseph Redmon e Ali Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016. arXiv: `1612.08242 [cs.CV]` (cit. a p. 32).

[34] Denny Vrandečić e Markus Krötzsch. «Wikidata: A Free Collaborative Knowledgebase». In: *Commun. ACM* 57.10 (set. 2014), pp. 78–85. ISSN: 0001-0782. DOI: `10.1145/2629489`. URL: `https://doi.org/10.1145/2629489` (cit. a p. 33).

[35] Steven Bird, Edward Loper e Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009 (cit. alle pp. 39, 53).

[36] Princeton University. *"About WordNet."* `https://wordnet.princeton.edu/`. 2010 (cit. alle pp. 39, 53).

[37] Srinivas Chakravarthy. *Tokenization for Natural Language Processing*. `https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4`. 2020 (cit. a p. 39).

[38] Lorenzo Canale, Laura Farinetti e Luca Cagliero. «From teaching books to educational videos and vice versa: a cross-media content retrieval experience». In: *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*. 2021, pp. 115–120. DOI: `10.1109/COMPSAC51774.2021.00027` (cit. alle pp. 42, 46, 73).

[39] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: `1405.0312 [cs.CV]` (cit. a p. 43).

[40] Moses e John Olafenwa. *ImageAI, an open source python library built to empower developers to build applications and systems with self-contained Computer Vision capabilities*. Mar. 2018. URL: `https://github.com/OlafenwaMoses/ImageAI` (cit. a p. 43).

[41] Jun Xu, Tao Mei, Ting Yao e Yong Rui. «MSR-VTT: A Large Video Description Dataset for Bridging Video and Language». In: IEEE International Conference on Computer Vision e Pattern Recognition (CVPR), giu. 2016. URL: https://www.microsoft.com/en-us/research/publication/msr-vtt-a-large-video-description-dataset-for-bridging-video-and-language/ (cit. alle pp. 57, 58).

[42] Ren Jie Tan. *Breaking Down Mean Average Precision (mAP)*. https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52. 2019 (cit. a p. 59).

[43] Michael Wray, Hazel Doughty e Dima Damen. *On Semantic Similarity in Video Retrieval*. 2021. DOI: 10.48550/ARXIV.2103.10095. URL: https://arxiv.org/abs/2103.10095 (cit. a p. 73).