

Practical
commonsense
reasoning

Ora che conosciamo la struttura di ConceptNet studiamo
come utilizzare questo strumento per fare qualche inferenza.

Identificare il contesto

Mentre la logica è microscopica, a grana fine, ben definita e statica, il contesto è macroscopico, gestalt (i.e., viene percepito come un tutt'uno), euristico e abbastanza dinamico.

ConceptNet eccelle nella risoluzioni di problemi legati al contesto proprio per le sue caratteristiche naturalmente dinamiche:

I creatori di ConceptNet hanno investito molto nel rappresentare come diversi concetti di common sense interagiscano fra loro anziché concentrarsi ossessivamente sulla ricerca della verità assoluta di alcune asserzioni

Dato un certo concetto, mediante le API di ConceptNet possiamo effettuare la chiamata `getContext()` la quale ci fornirà tutte le adiacenze e dunque ci aiuterà a distinguere in quale contesto il concetto è immerso

Es. se la query è **carrot** potremo trovare una relazione con *vegetable* e in questo modo comprendere che il contesto è quello delle verdure

Nota:
Chiaramente questa chiamata funziona bene su query poco o per nulla polisemiche

I risultato della chiamata `getContext()` è basata sulla misura della **relatedness**:

La relatedness di un certo nodo non è soltanto calcolata calcolando la distanza fra il concetto-query e quello restituito ma considera anche il numero e la forza dei path che collegano i due concetti.

In definitiva, la funzione `getContext()` è utile per l'espansione semantica (i.e., comprendere meglio l'input) e il topic generation:

Es. dando *restaurant* come input si otterrebbero frasi come *order food*, *water* e *menu* che potrebbero essere utilizzate in svariati modi per generare testo relativo all'input.

Analogy making

Per gli umani la creatività e le attività di apprendimento sono fortemente legate all'uso di analogie.

L'analogia è il rapporto di somiglianza tra alcuni elementi costitutivi di due fatti od oggetti, tale da far dedurre mentalmente un certo grado di somiglianza tra i fatti o gli oggetti stessi.

In ConceptNet due nodi sono analoghi se i loro set di archi entranti sono sovrapposti:

Es. giacché *apple* e *cherry* condividono gli stessi archi entranti [(PropertyOf x red); (PropertyOf x sweet); (IsA x fruit)] sono in qualche modo concetti analoghi.

Poter comprendere o produrre analogie/metafore apre una porta di fondamentale importanza sull'uso figurato del linguaggio, un aspetto che non abbiamo ancora incontrato e di cui parleremo in dettaglio in seguito.

Proiezioni

La proiezione è un cammino a partire da un nodo d'origine che è costituito sfruttando tutte e sole relazioni di un tipo che, però, deve essere transitivo (i.e., quel tipo di relazione che si sta utilizzando deve essere transitiva).

Es. Los Angeles è in California che è a sua volta negli USA: la relazione LocationOf è transitiva e dunque possiamo usare una proiezione e dire che Los Angeles è LocationOf USA senza dover esplicitamente indicare la relazione nel grafo.

Disambiguazione e classificazione

In ConceptNet è implementata una soluzione naive (i.e., primitivo) per la **classificazione** e la **disambiguazione**.

Dato un certo problema di classificazione/disambiguazione, per ogni classe/contesto si genera un prototipo (i.e., un documento d'esempio) il quale viene dato in input ad una funzione la quale computa le regioni di ConceptNet che quel certo esempio ricopre.

Dopodiché, ogni input fornito a ConceptNet viene associato ad una classe o disambiguato individuando l'esempio situato a distanza minima dalla zona ricoperta dall'input.

Questo sistema di classificazione è del tutto simile ad un approccio vettoriale statistico in cui si calcolano le distanze fra vettori di features (i.e., cosine similarity):

In questo caso però le feature sono basate su elementi della semantica common sense (e.g., tempo, spazio, sentimenti) piuttosto che basate feature basate sulla statistica (e.g., punteggiature, frequenza delle parole, ruoli semantici).

