Laboratorio di Intelligenza Artificiale: Ontologie e Ragionamento in Prolog con WordNet

Claudiu Daniel Hromei
05/12/2025

Introduzione

Oggi lavoreremo con Princeton WordNet, la mappa concettuale della lingua inglese più utilizzata al mondo. Il vostro obiettivo è programmare il cervello semantico di un agente AI, permettendogli di capire sinonimi, gerarchie e definizioni.

Strumenti Necessari:

- SWI-Prolog (ambiente desktop o SWISH online).
- File dati: *`mini_wordnet.pl`* (fornito dal docente) oppure il pacchetto completo.

Setup:

1. Create un nuovo file Prolog `lab_wordnet.pl`.
2. All'inizio del file, caricate il database con il comando: :- consult('mini_wordnet.pl').
3. Verificate il caricamento interrogando: ?- s(ID, _, 'dog', n, _, _). Dovreste ricevere degli ID numerici come risposta.

Modulo 1: Accesso Lessicale (Basic Queries)

Obiettivo: Navigare la distinzione tra parole (stringhe) e concetti (ID).

In WordNet, le parole sono ambigue. La parola "bank" può essere una banca (finanza) o una riva (fiume). WordNet disambigua assegnando a ogni significato un **SynsetID** diverso.

Fatti a disposizione:

- `s(ID, W_Num, 'parola', Tipo, Senso, Freq).`
- `g(ID, 'Definizione...').`

Esercizio 1.1: Il Cercatore di Sinonimi

Due parole sono sinonimi se condividono lo stesso SynsetID.

Scrivete un predicato `sinonimi(Parola, ListaSinonimi)` che, data una parola in input, restituisca una lista di tutte le parole alternative che hanno lo stesso significato.

Attenzione: Una parola può avere più sensi (più `ID`). Il vostro predicato deve gestire questa polisemia, magari restituendo più liste (una per ogni `ID`) o una lista unica appiattita.

Query di Test:

```
?- sinonimi('computer', L).
```

Atteso: `L = ['computing_machine', 'computing_device', 'data_processor',...]`

Esercizio 1.2: Glossario Semantico

Scrivete un predicato `spiega(Parola)` che stampi a video le definizioni di tutti i sensi possibili della parola, formattate in modo leggibile.

Suggerimento: Usate un ciclo fail o forall per stampare tutte le soluzioni, oppure il backtracking manuale della shell.

Modulo 2: Ragionamento Tassonomico (Gerarchie)

Obiettivo: Implementare la ricorsione su grafi.

La relazione `hyp(ChildID, ParentID)` definisce la gerarchia "is-a" (è un).

Esempio: `hyp(ID_Cane, ID_Mammifero).`

Esercizio 2.1: Iperonimo Diretto

Scrivete `genitore(Parola, Genitore)` che trova la categoria immediatamente superiore. Ricordate di passare attraverso gli ID:

```
Parola -> ID -> (hyp) -> ID_Genitore -> Parola_Genitore.
```

Esercizio 2.2: L'Antenato (Chiusura Transitiva)

Questa è la parte core dell'IA simbolica. Se X è un cane, e cane è un mammifero, e mammifero è un animale, allora X è un animale. Prolog non lo sa nativamente; dovete insegnarglielo.

Scrivete un predicato `iperonimo(Parola, Antenato)` che è vero se `Antenato` è un iperonimo diretto o indiretto di `Parola` a qualsiasi livello di profondità.

Hint: Questo richiede una regola ricorsiva.

1. Caso base: `A` è antenato di `B` se `hyp(B, A)`.
2. Passo ricorsivo: `A` è antenato di `B` se esiste un `X` tale che `hyp(B, X)` E `A` è antenato di `X`.

Domanda di Riflessione: Cosa succede se chiedete gli iperonimi di "`entity`" (la radice)? Come deve comportarsi il vostro codice?

Esercizio 2.3: Tassonomia Completa

Scrivete `catena_iperonimi(Parola, Lista)` che restituisca la lista ordinata dal concetto specifico fino alla radice.

Esempio: `[dog, canine, carnivore, placental, mammal, vertebrate, chordate, animal, organism, entity]`.

Modulo 3: Similarità e Inferenza

Obiettivo: Usare la struttura del grafo per calcolare metriche semantiche.

Esercizio 3.1: Fratelli Semantici (Co-Iponimia)

Due parole sono "fratelli" se hanno lo stesso genitore diretto (es. cane e gatto sono entrambi carnivori, o mammiferi, a seconda del livello).

Scrivete `fratello(A, B)` che verifica se due parole diverse condividono un genitore immediato.

Esercizio 3.2: Least Common Subsumer (LCS)

Il "Minimo Comune Iperonimo" è il concetto più specifico che include entrambi i termini. È fondamentale per calcolare quanto due parole sono simili.

- LCS(Gatto, Cane) = Carnivoro (molto simili).
- LCS(Gatto, Cucchiaio) = Entità fisica (molto distanti).

Scrivete `lcs(Parola1, Parola2, AntenatoComune)`:

1. Generate la catena degli antenati per `Parola1`.
2. Generate la catena degli antenati per `Parola2`.
3. Trovate il primo elemento che compare in entrambe le liste.

Modulo 4: Logica delle Azioni (Entailment) - *Avanzato*

Obiettivo: Usare `ent/2` per il ragionamento causale.

Usate il predicato `ent(ID_Verbo1, ID_Verbo2)` presente nel database (se incluso nel subset) o caricatelo. Questo predicato indica che fare l'azione 1 implica logica fare l'azione 2 (es. *Russare* implica *Dormire*).

Esercizio 4.1: Deduzione di Stato

Scrivete una regola `implica(AzioneA, AzioneB)` che, dato un verbo A (es. "snoring"), determini quali altre azioni sono logicamente vere (es. "sleeping").

Modulo 5: La Sfida della Disambiguazione (Complessità Combinatoria) - *Avanzato*

***Obiettivo:** Comprendere perché il linguaggio naturale è difficile per i computer.*

La maggior parte delle parole in WordNet è polisemica (ha più significati).

Se in una frase di 3 parole:

- La parola A ha 2 sensi;
- La parola B ha 5 sensi;
- La parola C ha 3 sensi;

Il computer deve valutare `2x5x3 = 30` possibili interpretazioni diverse di quella frase per capire quale sia quella corretta (Disambiguazione). Questo numero cresce esponenzialmente con la lunghezza della frase.

Esercizio 5.1: Il Calcolo Combinatorio

Scrivete un predicato `conta_interpretazioni(Frase, Totale)` che, data una frase (sotto forma di lista di parole), filtra e crea una lista di parole (soltanto quelle che sono presenti nel nostro WordNet) e infine calcoli quante combinazioni di sensi sono possibili.

Algoritmo suggerito:

1. Per ogni parola nella lista, contate quanti SynsetID (sensi) distinti esistono nel database (usando `findall` e `length`).
2. Moltiplicate questi conteggi tra loro (Accumulatore o ricorsione).Query di Test:

Provate con una delle frasi dello scenario, ad esempio:

1. *"The dog is a domestic animal."*
2. *"I parked my car near the vehicle."*
3. *"The robot has found a cup and a bottle in the kitchen."*
4. *"The human is snorling, so it's sleeping."*
5. *"The android perceives the sadness but has no emotions."*

Consegna

Consegnate il file `lab_wordnet.pl` che avete creato insieme ad un documento (Word o PDF) di spiegazione della vostra soluzione tramite email a:

- hromei@ing.uniroma2.it
- basili@info.uniroma2.it
- giacomo.deluca@hotmail.it