Relazione Esercizio 2

Nello sviluppo di questo esercizio, dopo una serie di prove abbiamo implementato due algoritmi, il primo, `genus_noun` che a partire dal Genus della definizione naviga verso il basso nell'albero di WordNet sfruttando gli iperonimi, e un secondo `genus_hyper` che sceglie come Genus l'iperonimo più frequente, che poi usa er navigare verso il basso l'albero di WordNet.a

La differenza tra i due algoritmi è minima (il secondo passaggio), per cui riportiamo entrambi gli pseudocodici evidenziano le parti differenti tra le due versioni.

genus_noun

1. Per ogni concetto (riga), esploriamo una definizione per volta (colonna).
2. Data la singola definizione, **prendiamo tutti i nomi** tramite un'analisi dei Pos Tag della frase. Fra questi **estraiamo quello più frequente** e lo **impostiamo come Genus** della definizione.
3. Per ogni definizione, ci salviamo la lista di iponimi calcolati a partire dal suo Genus.
4. Sfruttando l'oggetto CountVectorizer di Scikit-Learn otteniamo un vettore di frequenze degli iponimi presenti in tutte le definizioni relative allo stesso concetto.
5. Definiamo come concetto risultante l'iponimo più frequente tra tutti gli iponimi per quella definizione. In altre parole, il masimo elemento nel CountVectorizer.

genus_hyper

1. Per ogni concetto (riga), esploriamo una definizione per volta (colonna).
2. Data la singola **definizione preprocessata**, prendiamo **tutte le parole** che la compongono e le **disambiguiamo** una per una con **Lesk**. **Per ogni parola** disambiguata, **calcoliamo** i suoi **iperonimi** presenti in WordNet. Arriveremo ad ottenere una **lista di tutti gli iperonimi** per una **data definizione**. Da questa lista, **segliamo** come **Genus l'iperonimo più frequente**.
3. Per ogni definizione, ci salviamo la lista di iponimi calcolati a partire dal suo Genus.
4. Sfruttando l'oggetto CountVectorizer di Scikit-Learn otteniamo un vettore di frequenze degli iponimi presenti in tutte le definizioni relative allo stesso concetto.
5. Definiamo come concetto risultante l'iponimo più frequente tra tutti gli iponimi per quella definizione. In altre parole, il masimo elemento nel CountVectorizer.

Risultati

Riportiamo di seguito entrambi i risultati di entrambi gli algoritmi. Abbiamo eseguito diversi esperimenti a diverse profondità, a partire da profondità 1 (il primo iperonimo/iponimo) fino a profondità 20, dove sono aumentate di molto le tempistiche. A parità di livelli di esplorazione l'algoritmo `genus_hyper` si è dimostrato migliore nel calcolo del concetto finale, arrivando a dei risultati più vicini rispetto all'aloritmo `genus_noun`.

Depth - 1

```
Genus Noun (depth 1):
    1 - thing - 388
    2 - skill - 159
    3 - wish - 256
    4 - operation - 216
    5 - subject - 309
    6 - land - 141
    7 - land - 126
    8 - base_alloy - 35

Genus Hyper (depth 1):
    1 - theme - 43
    2 - focus - 16
    3 - cash - 16
    4 - network - 149
    5 - hard_time - 26
    6 - bus_company - 8
    7 - affine - 0
    8 - ride - 103
```

Depth - 2

```
Genus Noun (depth 2):
    1 - s_law - 982
    2 - capability - 67
    3 - wish - 736
    4 - s_law - 1120
    5 - water - 1136
    6 - case - 152
    7 - wall - 710
    8 - copper - 142

Genus Hyper (depth 2):
    1 - place - 406
    2 - break - 161
    3 - change - 57
    4 - accretion - 3
    5 - native - 370
    6 - electrical_system - 59
    7 - acanthuridae - 1
    8 - artificial_intelligence - 2
```

Depth - 3

```
Genus Noun (depth 3):
    1 - s_law - 1815
    2 - capability - 111
    3 - wish - 1217
```

```
    4 - s_law - 2491
    5 - thorax - 2421
    6 - body - 215
    7 - body - 140
    8 - chrome - 201

Genus Hyper (depth 3):
    1 - s_law - 216
    2 - break - 622
    3 - change - 36
    4 - accretion - 24
    5 - man - 1367
    6 - and - 45
    7 - accretion - 10
    8 - accretion - 9
```

Osservazioni

Abbiamo notato che scendendo troppo di livello (es.: 7, 10 o 20), entrambi gli algoritmi generalizzano troppo, tendendo a convergere verso pochi concetti (in partenza bisogna trovarne 8, a livello 20 si arriva magari a trovare 1 solo concetto per tutto). A tal proposito, nell'ottica di migliorare l'algoritmo abbiamo trovato molto difficile capire quando salire/scendere nell'albero di WordNet.

Sviluppi futuri

Come sviluppi futuri, si potrebbe implementare una terza versione dell'algoritmo, che anzichè restituire il genus più frequente, restituisse il genus che ha generato l'iponimo puù frequente, in modo da rendere più efficienti la ricerca e i risultati.