Relazione esercitazione – Hanks

L'obiettivo di questa esercitazione è stato quello di creare un algoritmo che implementasse la teoria di Patrick Hanks riguardante la costruzione del significato di frasi più o meno complesse. In particolare secondo Hanks, la radice del significato di una frase è da ritrovarsi nel verbo e nella sua valenza. Per poter capire quindi il significato di una frase occorre capire e studiare i diversi **semantic types** dei verbi che la compongono. Un verbo con valenza 2 avrà quindi 2 slots disponibili, e per ognuno di essi l'obiettivo sarà quello di studiare i fillers, ovvero i possibili termini che assumeranno un certo ruolo sintattico (ad es: soggetto e oggetto) in base allo slot in cui si troveranno.

Per questa esercitazione è stato utilizzato un corpus di frasi in lingua inglese, annotate tramite la tecnica del BIO-tagging, già utilizzato nel progetto fatto nella prima parte del corso con il professor Mazzei. Inizialmente il corpus era diviso in 3 parti: train, validation e test. Poiché, lo scopo per il quale è stato utilizzato in questa esercitazione è ben diverso, allora non c'è necessità di questa suddivisione e quindi tutte le 3 parti sono state considerate come un tutt'uno e successivamente non sono stati considerati i BIO-tags. Il corpus è presente nella cartella chiamata "**Risorse utili per esercitazione Hanks**" con il nome di "**datasets_en**".

I verbi transitivi considerati sono: **use** e **take**.

In tutte le frasi considerate i due verbi hanno valenza 2 e ai fini dell'applicazione della teoria di Hanks, sono stati individuati sostanzialmente 2 gruppi sintattici in riferimento ai 2 possibili argomenti del verbo:

- **POSSIBILI SOGGETTI**: subj, nsubjpass, nsubj.
- **POSSIBILI OGGETTI**: pobj, dobj, obj, iobj.

La decisione di considerare questi possibili soggetti e oggetti è stata presa affinchè l'algoritmo funzionasse correttamente e per essere sicuri che individuasse come fillers esclusivamente gli argomenti del verbo utilizzato.

Il programma sviluppato esegue i seguenti passi:

1) Per ogni verbo (use e love) vengono definite le declinazioni d'interesse che sono le seguenti: forma infinito (use,love) e terza persona singolare presente (uses,loves).
2) Vengono individuate le frasi del corpus in cui il verbo considerato in quel momento è presente in una delle declinazioni citate al passo 1).
3) Per ogni frase trovata viene calcolato il parser a dipendenze mediante l'utilizzo della libreria **spaCy** e successivamente vengono selezionate le parole presenti nel parser che rispettano i gruppi sintattici menzionati precedentemente e che sono praticamente gli argomenti del verbo. Tutto questo viene fatto dalla funzione chiamata "**parole_argomenti_verbo**".
4) Dopodichè, per ciascun filler trovato, viene eseguito l'algoritmo di Lesk tramite la chiamata alla funzione "**The_Lesk_Algorithm**". Questa implementazione da me realizzata, prende in input sia un argomento del verbo (ad esempio uno dei possibili soggetti oppure uno dei possibili oggetti) e sia tutta la frase corrente e restituisce il synset di Wordnet migliore da associare a quell'argomento avuto in input. Per fare questa disambiguazione l'algoritmo individua prima tutti i possibili synsets associati al termine di input e per ciascuno di essi considera tutti i suoi iperonimi e tutti i suoi iponimi. Come contesto di disambiguazione utilizza la frase corrente e come signature utilizza tutte le glosse e tutti gli

esempi sia del synset corrente e sia di tutti i suoi iperonimi e iponimi in modo da rendere più preciso il passo di disambiguazione. **Chiaramente, sia sul contesto che sulla signature, viene eseguita prima una fase di pre-processing** (eliminazione stop words, eliminazione eventuali stringhe vuote e lemmatizzazione) e solo dopo viene eseguita la WSD calcolando l'overlap tra il contesto e la signature mediante la funzione "**ComputeOverlap**"; questo viene fatto sempre per favorire maggiormente le sovrapposizioni lessicali.

5) Una volta ottenuti i due migliori synsets da associare ai due argomenti del verbo, sempre mediante l'ausilio di Wordnet, vengono individuati i loro rispettivi supersensi. I supersensi vengono restituiti dalla funzione chiamata "**supersensi**".

6) I supersensi ottenuti al passo precedente saranno i cosiddetti **semantic types dei due argomenti considerati in quel momento. Viene utilizzato un dizionario per memorizzare tutte le coppie di semantic types ottenute per ogni verbo e inoltre per ciascuna di esse viene memorizzata la sua frequenza. In questo modo, al termine dell'analisi di tutte le frasi di un certo verbo, ordinando tale dizionario in maniera decrescente si otterranno le coppie di semantic types per quel verbo da quella più ricorrente a quella meno ricorrente.**

Risultati ottenuti:

verbo –> use:

```
La coppia (communication-communication) è presente 25 volte su 397

La coppia (artifact-artifact) è presente 17 volte su 397

La coppia (person-communication) è presente 11 volte su 397

La coppia (communication-artifact) è presente 10 volte su 397

La coppia (group-communication) è presente 9 volte su 397

La coppia (person-artifact) è presente 9 volte su 397

La coppia (cognition-communication) è presente 8 volte su 397

La coppia (cognition-artifact) è presente 8 volte su 397

La coppia (location-artifact) è presente 8 volte su 397

La coppia (act-artifact) è presente 7 volte su 397
```

verbo -> take:

```
La coppia (act-contact) è presente 14 volte su 298

La coppia (communication-contact) è presente 10 volte su 298

La coppia (person-act) è presente 8 volte su 298

La coppia (artifact-contact) è presente 8 volte su 298

La coppia (event-contact) è presente 8 volte su 298

La coppia (person-artifact) è presente 7 volte su 298

La coppia (person-communication) è presente 6 volte su 298

La coppia (communication-communication) è presente 6 volte su 298

La coppia (person-group) è presente 6 volte su 298

La coppia (artifact-time) è presente 4 volte su 298
```