Descrizione delle modifiche all'algoritmo

L' algoritmo di Lesk per la word disambiguation prevede di fare una ricerca di tutti i possibili synsets della parola che si vuole disambiguare e filtrarli in base a quante parole della frase di input si ritrovano nelle definizioni dei diversi synset. Da questo algoritmo di partenza ho applicato 3 modifiche:

1) Non cerco tutti i synset ma solo quelli aventi lo stesso pos_tag della parola target. Solo in caso in cui non dovessi ottenere nessun risultato, li prenderei tutti in considerazione.

Esempio: nella frase "il punto chiave è il tempismo" prendo in considerazioni solo i synset che considerano chiave un aggettivo e non come un nome.

2) Prima di calcolare la sovrapposizione tra la frase e le varie definizioni, ne elimino la parola target contenuta in quanto non è significativo un match tra la parola e la frase che corrisponde alla sua stessa definizione.

Esempio: "La SOLUZIONE è facile, ci puoi arrivare."

Def.1 di soluzione = miscela omogenea di due o più sostanze, di solito una SOLUZIONE liquida

Def.2 di soluzione = sequenza di azioni necessarie per la risoluzione di un problema

3) L'algoritmo originale prevede di aggiornare il valore di best_sense solo se vi è un altro senso con un valore strettamente maggiori di overlap, ma non si cura di tutti i casi (frequenti) in cui vi sono più significati con lo stesso valore di overlap. Per questi casi ho pensato di utilizzare il concetto di concept similarity, quindi per ogni definizione:

A) effettuo del preprocessing sulla stessa: elimino tutte le stop words e lemmatizzo le parole rimanenti, trasformando la frase in una piccola lista di parole.
   a) **Esempio**: " a statement that solves a problem or explains how to solve the problem" diventa → {'explain', 'problem', 'statement', 'solve'}

B) calcolo l' overlap con la frase di input
C) nel caso in cui l' overlap sia pari a quello massimo attuale applico l'algoritmo di content similarity tra tutte le parole della definizione e quella target, data in input. Raccolgo tutti valori di cs calcolate in una somma e divido il risultato per il numero di parole della definizione (per evitare l'ingiusta vittoria di definizioni semplicemente più lunghe e non più attinenti)
D) il best-sense è quello che ha sia l' overlap che il CS maggiori

Esempio:
Frase: Work out the solution in your head.
Parola di disambiguare: solution

Prima Definizione: a homogeneous mixture of two or more substances; frequently (but not necessarily) a liquid
Definizione processata: {'liquid', 'frequently', 'homogeneous', 'necessarily', 'substance', 'mixture'}
Overlap: set() -> nessuna sovrapposizione

Cs Totale: 0.10337605300653312
Seconda Definizione: a statement that solves a problem or explains how to solve the problem
Definizione processata: {'explain', 'problem', 'statement', 'solve'}
Overlap: set()
Cs Totale: 0.16897865029528825

Definizione: a method for solving a problem
Definizione processata: {'problem', 'solve', 'method'}
Overlap: set()
Cs Totale: 0.27785784128355

Definizione: the set of values that give a true statement when substituted into an equation
Definizione processata: {'set', 'equation', 'true', 'statement', 'substitute', 'value'}
Overlap: set()
Cs Totale: 0.15931892010796092

Definizione: the successful action of solving a problem
Definizione processata: {'action', 'problem', 'successful', 'solve'}
Overlap: set()
Cs Totale: 0.22900936601204208

Notiamo come le definizioni dal CS più alto sono quelle aventi la parola “problem” la quale da sola garantisce un incremento sulla CS totale di 0.5837812889672553. (il valore finale viene diviso per il numero totale di parole della definizione). Inoltre in questo caso nessuna parola della frase ha matching con quelle della definizione in quanto la parola “solution” non viene presa in considerazione, altrimenti l’algoritmo avrebbe dato priorità al concetto di overlap e avrebbe selezionato la prima definizione (errata).

Pseudocodice

```
function CSLesk(word,sentence)
        all_tag = pos_tag(sentence)
        word_tag = all_tag[word]
        synsets = getSynset(word, word_tag) //estratto solo i synset con il tag di interesse.
        local_overlap = 0 //overlap del senso preso in esame
        global_overlap = 0 //l' overlap più grande tra tutti
        local_cs = 0 // in caso di pari overlap uso la content similarity per scegliere
        global_cs = 0
        for senses in synsets do
                definition = senses.getDefinition()
                definition = definition.remove(word) #elimino la parola dalla definizione
                overlap = ComputeOverlap(sentence,definition)
                if overlap > max-overlap then
                        max-overlap = overlap
                        best-sense = sense
                else
                        if local_overlap == max_overlap:
           local_cs = compute_cs(definition,word)
           if local_cs > max_cs:
              max_cs = local_cs
              best_sense = sense
        end for
        return best-sense
```

Risultati Esercitazione

Sentence: Arms bend at the elbow. -> Word: arms

My Lesk: a human limb; technically the part of the superior limb between the shoulder and the elbow but commonly used to refer to the whole superior limb (Synset('arm.n.01'))

Fifo Lesk: a human limb; technically the part of the superior limb between the shoulder and the elbow but commonly used to refer to the whole superior limb (Synset('arm.n.01'))

My CS Lesk: the part of an armchair or sofa that supports the elbow and forearm of a seated person (Synset('arm.n.04'))

Wordnet Library Lesk: the part of a garment that is attached at the armhole and that provides a cloth covering for the arm (Synset('sleeve.n.01'))

Sentence: Germany sells arms to Saudi Arabia. -> Word: arms

My Lesk: any projection that is thought to resemble a human arm (Synset('arm.n.02'))

Fifo Lesk: a human limb; technically the part of the superior limb between the shoulder and the elbow but commonly used to refer to the whole superior limb (Synset('arm.n.01'))

My CS Lesk: any instrument or instrumentality used in fighting or hunting (Synset('weapon.n.01'))

Wordnet Library Lesk: supply with arms (Synset('arm.v.02'))

Sentence: The key broke in the lock. -> Word: key

My Lesk: metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated (Synset('key.n.01'))

Fifo Lesk: metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated (Synset('key.n.01'))

My CS Lesk: metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated (Synset('key.n.01'))

Wordnet Library Lesk: metal device shaped in such a way that when it is inserted into the appropriate lock the lock's mechanism can be rotated (Synset('key.n.01'))

Sentence: The key problem was not one of quality but of quantity. -> Word: key

My Lesk: serving as an essential component (Synset('cardinal.s.01'))

Fifo Lesk: serving as an essential component (Synset('cardinal.s.01'))

My CS Lesk: serving as an essential component (Synset('cardinal.s.01'))

Wordnet Library Lesk: United States lawyer and poet who wrote a poem after witnessing the British attack on Baltimore during the War of 1812; the poem was later set to music and entitled `The Star-Spangled Banner' (1779-1843) (Synset('key.n.07'))

Sentence: Work out the solution in your head. -> Word: solution

My Lesk: a homogeneous mixture of two or more substances; frequently (but not necessarily) a liquid solution (Synset('solution.n.01'))

Fifo Lesk: a homogeneous mixture of two or more substances; frequently (but not necessarily) a liquid solution (Synset('solution.n.01'))

My CS Lesk: a method for solving a problem (Synset('solution.n.03'))

Wordnet Library Lesk: the successful action of solving a problem (Synset('solution.n.05'))

Sentence: Heat the solution to 75° Celsius. -> Word: solution

My Lesk: a homogeneous mixture of two or more substances; frequently (but not necessarily) a liquid solution (Synset('solution.n.01'))

Fifo Lesk: a homogeneous mixture of two or more substances; frequently (but not necessarily) a liquid solution (Synset('solution.n.01'))

My CS Lesk: a method for solving a problem (Synset('solution.n.03'))

Wordnet Library Lesk: a statement that solves a problem or explains how to solve the problem (Synset('solution.n.02'))

Sentence: The house was burnt to ashes while the owner returned. -> Word: ashes

My Lesk: the residue that remains when something is burned (Synset('ash.n.01'))

Fifo Lesk: the residue that remains when something is burned (Synset('ash.n.01'))

My CS Lesk: the residue that remains when something is burned (Synset('ash.n.01'))

Wordnet Library Lesk: convert into ashes (Synset('ash.v.01'))

Sentence: This table is made of ash wood. -> Word: ash

My Lesk: strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats (Synset('ash.n.03'))

Fifo Lesk: strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats (Synset('ash.n.03'))

My CS Lesk: strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats (Synset('ash.n.03'))

Wordnet Library Lesk: strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats (Synset('ash.n.03'))

Sentence: The lunch with her boss took longer than she expected. -> Word: lunch

My Lesk: a midday meal (Synset('lunch.n.01'))

Fifo Lesk: a midday meal (Synset('lunch.n.01'))

My CS Lesk: a midday meal (Synset('lunch.n.01'))

Wordnet Library Lesk: provide a midday meal for (Synset('lunch.v.02'))

Sentence: She packed her lunch in her purse. -> Word: lunch

My Lesk: a midday meal (Synset('lunch.n.01'))

Fifo Lesk: a midday meal (Synset('lunch.n.01'))

My CS Lesk: a midday meal (Synset('lunch.n.01'))

Wordnet Library Lesk: provide a midday meal for (Synset('lunch.v.02'))

Sentence: The classification of the genetic data took two years. -> Word: classification

My Lesk: the act of distributing things into classes or categories of the same type (Synset('categorization.n.03'))

Fifo Lesk: the act of distributing things into classes or categories of the same type (Synset('categorization.n.03'))

My CS Lesk: a group of people or things arranged by class or category (Synset('classification.n.02'))

Wordnet Library Lesk: the basic cognitive process of arranging into classes or categories (Synset('classification.n.03'))

Sentence: The journal Science published the classification this month. -> Word: classification

My Lesk: the act of distributing things into classes or categories of the same type (Synset('categorization.n.03'))

Fifo Lesk: the act of distributing things into classes or categories of the same type (Synset('categorization.n.03'))

My CS Lesk: a group of people or things arranged by class or category (Synset('classification.n.02'))

Wordnet Library Lesk: restriction imposed by the government on documents or weapons that are available only to certain authorized people (Synset('classification.n.04'))

Sentence: His cottage is near a small wood. -> Word: wood

My Lesk: the hard fibrous lignified substance under the bark of trees (Synset('wood.n.01'))

Fifo Lesk: the hard fibrous lignified substance under the bark of trees (Synset('wood.n.01'))

My CS Lesk: any wind instrument other than the brass instruments (Synset('woodwind.n.01'))

Wordnet Library Lesk: a golf club with a long shaft used to hit long shots; originally made with a wooden head (Synset('wood.n.08'))

Sentence: The statue was made out of a block of wood -> Word: wood

My Lesk: the hard fibrous lignified substance under the bark of trees (Synset('wood.n.01'))

Fifo Lesk: the hard fibrous lignified substance under the bark of trees (Synset('wood.n.01'))

My CS Lesk: any wind instrument other than the brass instruments (Synset('woodwind.n.01'))

Wordnet Library Lesk: a golf club with a long shaft used to hit long shots; originally made with a wooden head (Synset('wood.n.08'))