

Relazione esercizio 2: edit distance

Studente: Cagnazzo Christian Damiano – Matricola: 883100

La prima versione dell'algoritmo edit distance ha un tempo di esecuzione molto alto a causa delle numerose chiamate ricorsive sui sotto problemi, che spesso sono ripetuti.

Sappiamo che date due stringhe $S1$ e $S2$ se $S1[i]=S2[j]$ l'edit distance tra $S1[1...i]$ e $S2[1...j]$ sarà uguale a quella tra $S1[1...i-1]$ e $S2[1...j-1]$ (in quanto non bisogna fare modifiche), al contrario se $S1[i] \neq S2[j]$ l'edit distance tra $S1[1...i]$ e $S2[1...j]$ sarà uguale al minimo tra 1 più l'edit distance tra $S1[1...i]$ e $S2[1...j-1]$ e 1 più l'edit distance tra $S1[1...i-1]$ e $S2[1...j]$. Ovviamente, se una delle due stringhe è vuota l'edit distance è uguale alla lunghezza dell'altra stringa.

Possiamo sfruttare ciò pensando di migliorare l'algoritmo edit distance e in più salvare man mano i risultati dei sotto problemi, affinché questi non debbano essere ogni volta risolti, sfruttando le tecniche della programmazione dinamica.

Per salvare i risultati possiamo definire quindi una matrice T della dimensione $[lunghezza_prima_stringa \times lunghezza_seconda_stringa]$ in cui nelle posizioni $T[i,j]$ si trova l'edit distance tra le stringhe $S1[1..i]$ e $S2[1..j]$. $T[i,j]$ è così definita:

- i se $j < 0$ e $i > 0$ (S2 è vuota)
- j se $i < 0$ e $j > 0$ (S1 è vuota)
- 0 se i e j sono < 0 (S1 e S2 vuote)
- $T[i-1, j-1]$ se $S1[i] = S2[j]$
- $\min(1+T[i, j-1], 1+T[i-1, j])$ se $S1[i] \neq S2[j]$

Eseguito degli esperimenti sull'applicazione e testando il nuovo algoritmo si è osservato che calcolando l'edit distance della prima parola del testo ("Quando") con le sole prime 5 parole del dizionario si ottengono in totale 60 accessi alla tabella creata, che corrispondono quindi all'aver evitato di effettuare 60 chiamate ricorsive su sotto-problemi che avevamo già calcolato. Gli accessi diventano 135 con le prime 10 parole del dizionario e 266 con le prime 20. E' facile immaginare quanti accessi alla tabella, e quindi quanti sotto-problemi, si riescano ad evitare calcolando l'edit distance tra tutte le parole del dizionario (circa 660 mila parole) e ogni parola del testo (circa 50). L'algoritmo, infatti, impiega meno di un secondo per calcolare l'edit distance tra una parola del testo e tutte le parole del dizionario. In totale, ci vogliono circa 30 secondi per calcolare l'edit distance di tutte le parole del testo tra quelle del dizionario e per associare ad ogni parola del testo una lista delle parole del dizionario con l'edit distance minimo.