EditDistance e indici n-gram

Luca Maltempo

October 2021

1 Introduzione

L'algoritmo Edit Distance permette di calcolare la distanza tra due parole di un dizionario, questo algoritmo, implementato attraverso diverse tecniche, viene usato da tutti i programmi di scrittura, in modo da riconoscere eventuali errori durante la digitazione del testo e correggerli consigliando le parole del dizionario che sono più vicine a quella che è stata digitata. In generale vogliamo che l'operazione di calcolo della distanza tra una parola e le restanti del dizionario sia molto veloce, questo perchè abbiamo bisogno che la correzione della parola errata sia fatta in tempo reale.

2 Edit Distance

L'algoritmo riceve due stringhe di caratteri X e Y e calcola la loro distanza di edit che è pari al minimo numero di operazioni elementari necessarie per far in modo che le due stringhe abbiano lo stesso contenuto. Ogni operazione ha un costo che poi determina la distanza effettiva tra le due parole.

Operazioni elementati e costo:

- Copia, 0 = Lascia invariato un carattere e ha costo nullo.
- Sostituzione, 1 = Sostituisce un carattere della prima stringa col carattere corrispondente della seconda stringa.
- Scambio, 1 = Scambia due caratteri di una stringa se questi risultano in posizioni invertite nella seconda stringa.
- Cancellazione, 1 = Cancella un carattere da una stringa.
- Inserimento, 1 = Inserisce un carattere in una stringa.

Come per il problema della massima sottosequenza comune (LCS), anche in questo caso è possibile definire una sottostruttura ottima valutando le due stringhe un carattere alla volta. Sia m la lunghezza di X e n la lunghezza di Y, definiamo $X_i =< x_0, x_1, ..., x_i >$ e $Y_j =< y_0, y_1, ..., y_j >$ sottosequenze di

X e Y. I sottoproblemi da risolvere sono il calcolo della distanza di edit sulle sottosequenze X_i e Y_j per i, j = 0, 1,..., m, n. Una soluzione ottima al problema $X_i \rightarrow Y_j$ deve includere una soluzione ottima al problema $X_{i-1} \rightarrow Y_{j-1}$. Definisco C[i][j] il costo di una soluzione ottima al problema $X_i \rightarrow Y_j$ nel seguente modo: C[i][j] = C[i-1][j-1] + costo(operazione). Attraverso una formulazione ricorsiva arriviamo alla risoluzione del problema di Edit-Distance in un tempo pari a $\theta(m \cdot n)$

3 Utlizzare l'Edit Distance

L'edit distance per la ricerca della parola più vicina ad una query può essere implementata in diversi modi. La tecnica più naive vede l'esecuzione dell'algoritmo su tutte le parole del dizionario. Il dizionario può essere più o meno grande ma in generale fare l'edit distance sull'intero dizionario non è mai consigliabile. Per diminuire i tempi di esecuzione dell'algoritmo si potrebbe fare l'edit distance solo con le parole che hanno una lunghezza simile alla query, mantenendo così un buon hit rate ma il tempo di esecuzione, nel caso in cui il dizionario fosse grande, resterebbe proibitivo. Per questo si fa affidamento agli indici n-gram.

3.1 N-gram

L' n-gram di una parola corrisponde all'insieme di tutte le sottosequenze di lunghezza n della parola. La creazione degli indici n-gram di tutte le parole del dizionario permette un utilizzo più furbo dell'edit-distance. Infatti, invece di eseguire il calcolo della distanza tra la query e tutte le parole del dizionario, l'edit distance è eseguita solo nel caso in cui le due parole abbiano un numero sufficientemente alto di sottosequenze in comune. Date due stringhe, definisco con X e Y i rispettivi insiemi di sottosequenze di lunghezza n, e ricavo il Coefficiente di Jaccard tramite questa semplice formula: J = $\frac{X \cap Y}{X \cup Y}$. L'edit Distance tra le due parole viene realizzato solo se il coefficiente di Jaccard è superiore ad un valore da noi scelto. Maggiore è questa soglia e minore sarà il numero di edit distance che vengono eseguiti, in questo modo si riesce a diminuire in modo considerevole il tempo di esecuzione dell'algoritmo ma si va anche a diminuire il suo Hit Rate.

4 Esperimenti svolti

Per testare l' efficacia dei diversi algoritmi implementati nel codice è stato utilizzato un dizionario contenente 95 mila parole italiane, tra queste sono presenti anche nomi propri e località. Ovviamente, il tempo di esecuzione dell' Edit Distance varia molto in base alla lunghezza del dizionario utilizzato oltre che alla tecnica scelta per l'implementazione dell'algoritmo. Per reallizare i vari test vengono scelte casualmente 100 parole dal dizionario, e su di queste viene effettuato un numero di operazioni elementari. Questo numero è scelto casualmente

tra 1 e la parte inferiore di metà della lunghezza della parola. Per ogni operazione elementare da effettuare su una parola si sceglie, sempre casualmente, una posizione nella quale inserire un carattere, cancellare il carattere presente, sostituirlo con un altro dell'alfabeto o scambiarlo col carattere precedente. In questo modo si avranno due liste di 100 parole, la prima contenente le parole originarie e la seconda contenente le parole modificate che andranno a simulare eventuali errori di battitura. A questo punto non rimane altro che calcolare l'edit distance su ogni parola modificata e vedere se la parola ad essa più vicina corrisponde a quella originaria, Hit, o meno, Miss. Per l'edit distance sono state utilizzate 3 diverse modalità, la versione naive prevede l'edit distance tra la parola modificata e l'intero dizionario. La seconda versione esegue l'edit distance tra la parola modificata e una parola del dizionario solo se la differenza della lunghezza delle due parole è minore o uguale ad uno. Infine l'ultima versione sfrutta gli indici n-gram, per la precisione 2-gram, 3-gram e 4-gram, e realizza l'edit distance tra la parola modificata e una del dizionario solo se il loro coefficiente di Jaccard è maggiore della soglia indicata. Per l'esperimento sono state scelte tre soglie: 0.2, 0.4 e 0.8 . Visto che ogni parola nel dizionario può essere trovata sia al singolare che al plurale ho riportato più risultati per quanto riguarda l'Hit rate. Prima viene calcolato tenendo conto solo della parola più vicina alla query e poi valutando se la parola cercata corrisponde ad una delle prime due parole a distanza minore dalla query. In un'eventuale applicazione, ad esempio la tastiera dello smartphone, è infatti comune che vengano suggerite due parole vicine a quella che viene scritta e non solo una.

5 Risultati sperimentali

L'Edit Distance con tutte le parole del dizionario ha un Hit rate pari a 0.67 e un tempo medio di esecuzione per ogni parola di 85 secondi.
L'Edit Distance solo con le parole di lunghezza simile ha un Hit rate pari a 0.36 e un tempo medio di esecuzione per ogni parola di 24 secondi.
In tabella sono riportati i risultati dell' Edit Distance attraverso indici n-Gram. Le prime due tabelle fanno riferimento al caso in cui l'Hit rate sia calcolato rispetto alla parola più vicina o al caso in cui si tenga conto delle prime due parole a distanza minore. La terza tabella indica invece il tempo medio di esecuzione dell'algoritmo per una parola in secondi. Ovviamente il tempo impiegato è lo stesso indipendentemente dal modo con cui si scelga di calcolare l'hit rate.

Hit Rate Edit Distance			
Indici n-Gram	Jaccard Treshold >= 0.2	Jaccard Treshold >= 0.4	Jaccard Treshold >= 0.8
2-Gram	0.65	0.55	0.06
3-Gram	0.56	0.34	0.02
4-Gram	0.39	0.24	0.02

Hit Rate Edit Distance			
Indici n-Gram	Jaccard Treshold >= 0.2	Jaccard Treshold >= 0.4	Jaccard Treshold >= 0.8
2-Gram	0.70	0.60	0.06
3-Gram	0.61	0.35	0.02
4-Gram	0.40	0.25	0.02

Tempo Edit Distance in secondi			
Indici n-Gram	Jaccard Treshold >= 0.2	Jaccard Treshold >= 0.4	Jaccard Treshold >= 0.8
2-Gram	1.52	0.39	0.37
3-Gram	0.43	0.33	0.32
4-Gram	0.31	0.28	0.28

6 Conclusioni

Si intuisce immediatamente che, per applicazioni real time, le implementazioni classiche dell'Edit distance non siano utilizzabili. Sfuttare gli indici n-Gram permette invece di ridurre in maniera considerevole il tempo necessario per l'esecuzione dell'algoritmo. Dai risultati è evidente che le combinazioni migliori per ottenere un Hit Rate elevato e un breve tempo di esecuzione siano solo due, 3-Gram con soglia di Jaccard >= 0.2 o indici 2-Gram con soglia di Jaccard >= 0.4 . Attraverso entrambe le combinazioni è possibile ottenere un tempo di esecuzione minore di mezzo secondo e un Hit Rate maggiore del 55%, inoltre a parità di indice n-Gram e soglia di Jaccard, l'Hit Rate aumenta se si tengono in considerazione le prime due parole più vicine alla query senza che il tempo di esecuzione dell'algoritmo subisca variazioni.