CIRT LAVINIA
FAGIOLI PAOLO
KODRA KRISTIAN

RELAZIONE: APPLICAZIONE RECTIFY

Abbiamo sviluppato due versioni dell'applicazione Rectify (che utilizza la funzione edit_distance_dyn), la prima versione ha impiegato intorno ai 10 minuti per completare l'esecuzione.
L'idea era quella di scorrere tutto il dizionario per ogni parola della frase , e inserire nell'ArrayList le parole con edit_distance minimo. Nel caso poi si trovasse la parola uguale nel dizionario, questa sarebbe stata selezionata.
Però il tempo richiesto dall'algoritmo per completare l'esecuzione ci è sembrato irragionevole.
Abbiamo cercato di migliorarlo nella seconda versione in cui abbiamo cambiato il modo di approcciarci al problema, poiché ci siamo accorti che stavamo calcolando l'edit_distance in partenza, anche delle parole presenti nel dizionario (es. con la parola "quando"calcolavamo l'edit_distance per tutte le parole nel dizionario dalla a alla q), abbiamo preferito, invece di aprire il dizionario una sola volta e calcolare l'edit_distance minimo a prescindere, aprire il dizionario una prima volta per controllare se la parola del file "correctme.txt" fosse presente all'interno, se positivo questa sarebbe stata stampata direttamente senza calcolare l'edit_distance minimo (uno spreco di tempo risparmiabile). Nel caso l'esito del primo controllo del dizionario fosse risultato negativo, sarebbe stato necessario riaprire il dizionario per calcolare l'edit_distance minimo. Attraverso questa strategia siamo riusciti a dimezzare il tempo di esecuzione richiesto dall'algoritmo.

Abbiamo notato che alcune parole non venivano corrette, ad esempio la parola "selice" che, da nostra idea, sarebbe dovuta essere corretta in "felice". Poiché presente nel dizionario, essa non viene modificata, perché non riconosciuta come errore.
Abbiamo constatato che il modo di correggere da parte dell'applicazione è differente da quello umano e non tutte le correzioni vengono eseguite in maniera logica.

Per scrupolosità abbiamo provato ad utilizzare nell'applicazione la versione non dinamica del metodo edit_distance, per verificare che effettivamente la versione dinamica fosse più veloce , il tempo di esecuzione è apparso interminabile, dato che ricalcolava anche i sottoproblemi comuni. La risposta di questa versione è stata peggiore sia della prima che della seconda versione sviluppata quindi ha confermato le nostre teorie.