RELAZIONE ESERCIZIO 2

La classe EditDistanceUsage utilizza la funzione edit_distance_dyn per determinare, per ogni parola w in correctme.txt, la lista di parole in dictionary.txt con edit distance minima da w.

I primi due metodi della classe , loadFile e loadDictionary, caricano le parole dei file correctme.txt e dictionary.txt in due ArrayList distinti: correctme per il file correctme.txt e dictionary per il file dictionary.txt.

Il confronto tra le due liste viene fatto con la funzione corrector.
Corrector prende in input i due ArrayList e nel primo ciclo foreach ne crea un altro, minWords, in cui verranno salvate le parole di dictionary.txt con edit distance minima da w.
Istanzia inoltre la variabile min, in cui memorizza il più grande valore rappresentabile di tipo intero.

Nel secondo foreach il metodo scorre l'ArrayList dictionary la funzione edit_distance_dyn calcola la distanza tra le stringhe d e w , prese rispettivamente da dictionary e correctme.
Il risultato viene poi memorizzato nella variabile dist, ed aggiunto all'ArrayList minWords
nel caso in cui dist fosse minore o uguale alla variabile min.

In seguito vengono stampate tutte le parole seguite dalle possibili correzioni.
Sperimentando l'utilizzo dell'applicazione, abbiamo notato che le parole seguite da un carattere di punteggiatura vengono considerate come parole "errate", pertanto la funzione setaccia il dizionario per correggerla.
A livello di prestazioni l'applicazione ha un tempo di esecuzione di circa 26 secondi.
La complessità dell'algoritmo edit_distance_dyn è O(mn), pertanto la complessità lineare garantisce un buon tempo di esecuzione.
Il caso peggiore si ha quando la dimensione del file da correggere eguaglia quella del dizionario; in questo caso la complessità diventa O(n^2).