RELAZIONE ESERCIZIO 2

Per l'esercizio sono stati implementati due versioni dell'algorimto "Edit Distance" o anche chiamato "Levenshtein Distance" , date due parole ne calcola il numero minimo di operazioni ammesse (in questo caso: inserimento, cancellazione e sostituzione) per trasformare la prima parola nella seconda.

Entrambe le versioni dell'algoritmo utilizzano la ricorsione ma solo la seconda versione è stata progettata con tecniche di programmazione dinamica.

l'applicazione utilizza la versione dell'algoritmo implementato con programmazione dinamica, grazie all'uso di questa tecnica confrontando l'esecuzione dei due algoritmi si può notare una netta differenza in termini di tempi di esecuzione.

Per evitare di confrontare ogni parola presente in correctMe.txt per ogni parola in dictionary.txt, le parole nel dizionario sono state ordinate per lunghezza(crescente), grazie a questa strategia quando si trova l'edit distance minima e la parola successiva da confrontare è maggiore rispetto a questo valore, l'algoritmo blocca il confronto e passa alla parola successiva da correggere in correctMe.txt. Questo permette un ottimizzazione dell'algoritmo in cui solo nel caso peggiore viene scansionato l'intero dizionario.

Dai risultati ottenuti dall'output dell'algoritmo possiamo osservare che data una parola in correctMe.txt le possibili correzioni possono essere svariate, l'algoritmo non è in grado di trovare la parola corretta in base al contesto.

```
Word: perpeteva Edit distance: 2

perpetra | perpetua | repeteva | percoteva | perteneva | permetteva | perpendeva |
perpetrava | perpetrera | perpetuava | perpetuera
```

Andrea Azzalin
778097