



Esercizio numero 2

Tiziano Jhonny Floriddia , 1041726

Nicolas Fiocco, 1052526

Giorgio Francesco Floridia , 1054584

10/12/2024

RELAZIONE: ALGORITMO DI EDIT DISTANCE

Panoramica

Questa relazione descrive un'applicazione che calcola la distanza di edit minima tra stringhe utilizzando la funzione `'edit_distance_dyn'`. La distanza di edit rappresenta il numero minimo di operazioni (inserimenti, cancellazioni) necessarie per trasformare una stringa in un'altra. Esempi:

- "casa" → "cassa" ha una distanza di edit pari a 1 (1 inserimento).
- "salsa" → "salse" ha una distanza di edit pari a 2 (1 cancellazione e 1 inserimento).
- "motore" → "moto" ha una distanza di edit pari a 2 (2 cancellazioni).
- "tempo" → "sempre" ha una distanza di edit pari a 4 (3 cancellazioni e 1 inserimento).
- "trave" → "trave" ha una distanza di edit pari a 0 (nessuna modifica).

Dettagli dell'implementazione

L'applicazione si articola nei seguenti passaggi principali:

1. **Lettura dell'input:** Legge il file `'correctme.txt'` contenente le parole da analizzare.
2. **Confronto con il dizionario:** Confronta ogni parola in `'correctme.txt'` con quelle presenti in un vocabolario di riferimento `'dictionary.txt'`.
3. **Individuazione delle corrispondenze:** Per ogni parola in `'correctme.txt'`, restituisce l'elenco delle parole del dizionario con distanza di edit minima. Se la parola è già presente nel dizionario, viene restituita con una distanza di edit pari a 0.

Per garantire coerenza nei confronti, la funzione `'remove_punctuation'` normalizza le parole tramite la rimozione di segni di punteggiatura.

Esempi:

- `"vita."` diventa `"vita"` (distanza di edit 0).

Analisi delle prestazioni

L'algoritmo presenta un comportamento prevedibile: i tempi di esecuzione aumentano in modo approssimativamente lineare con il numero di parole da confrontare. I fattori che influenzano le prestazioni includono la dimensione e la struttura del dizionario e la complessità delle parole di input. Gli esperimenti condotti confermano che l'algoritmo è adatto a compiti di correzione ortografica in tempo reale.

Limitazioni e osservazioni

- **Efficienza:** L'algoritmo è efficace nell'identificare errori ortografici ma non è progettato per rilevare errori semantici, infatti fornisce un elenco di parole simili, che però non garantiscono di mantenere il significato originale nel contesto della frase.