文字列の類似度評価

2つの文字列がどのぐらい似ているかということを調べる手順はどう定義すれば良い？

文字列の類似度評価

2つの文字列がどのぐらい似ているかということを調べる手順はどう定義すれば良い？

A）一致している文字の数を数える

文字列の類似度評価

2つの文字列がどのぐらい似ているかということを調べる手順はどう定義すれば良い？

A）一致している文字の数を数える

弱点：文字は同じだが順番が異なるケース

例：きつそう
うそつき
➔ 100%一致

文字列の類似度評価

2つの文字列がどのぐらい似ているかということを調べる手順はどう定義すれば良い？

A）一致している文字の数を数える

B）順番に文字を比べて、一致している数を数える

例：きつそう
うそつき
→ 0%一致

文字列の類似度評価

2つの文字列がどのぐらい似ているかということを調べる手順はどう定義すれば良い？

A）一致している文字の数を数える

B）順番に文字を比べて、一致している数を数える

例１：きつそう
うそつき
→ 0%一致

例２：おつかれさん
つかれました
→ 0%一致

文字列の類似度評価

実際に自然言語処理研究で使われる方法：

- 「**編集距離**」または「**レーベンシュタイン距離**」（Levenshtein distance）と呼ばれる
- 一致している文字数ではなく、二つの文字列がどの程度異なっているか調べる
- 1つの文字列をもう一つの文字列に変換するに必要な1文字の変更処理を数える

  - 挿入
  - 削除
  - 置換

例：おつかれさん
つかれました → 4

レーベンシュタイン距離（Levenshtein distance）

レーベンシュタイン距離を計算してみよう:

k	i	t	t	e	n	
s	i	t	t	i	n	g

計算の手順は説明できる？