Un correcteur orthographique et syntaxique

Correcteur orthographique et syntaxique

I. Distance entre deux mots

II. Deux algorithmes de correction orthographique
   A. Méthode des trigrammes
   B. Méthode probabiliste

III. Algorithme final

I. Distance entre deux mots

- Exemples de distances entre deux chaînes de caractère
  - Distance de Damerau-Levenshtein
    - Nombre minimum d'opérations pour passer d'une chaîne à l'autre (insertion, suppression, substitution d'un caractère ou transposition de deux caractères adjacents)

I. Distance entre deux mots

- Exemples de distances entre deux chaînes de caractère
  - Distance de Damerau-Levenshtein
  - Distance de Jaccard
    - Soient $A = a_1 a_2 \dots a_n \; et \; B = b_1 b_2 \dots b_p$

$$J(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

I. Distance entre deux mots

- Exemples de distances entre deux chaînes de caractère
  - Distance de Damerau-Levenshtein
  - Distance de Jaccard
  - Distances plus particulières (dactylographique, phonétique…)

II. Deux algorithmes de correction orthographique

- Comparaison mot-à-mot irréalisable
  - ➔ Nécessité de comparer avec un dictionnaire plus réduit
- Méthode des trigrammes
- Méthode probabiliste

II. A) Méthode des trigrammes

1) `"test"` → `["$te", "tes", "est", "st$"]`

```
>>> dictionnaire_trigrammes["est"]
["test", "estuaire", "peste", "est", …]
```

II. A) Méthode des trigrammes

1) `"test"` → `["$te", "tes", "est", "st$"]`

```
>>> dictionnaire_trigrammes["est"]
["test", "estuaire", "peste", "est", …]
```

2) ▪ Recherche des mots ayant le plus de trigrammes en commun avec le mot à corriger

II. A) Méthode des trigrammes

1) `"test"` → `["$te", "tes", "est", "st$"]`

```
>>> dictionnaire_trigrammes["est"]
["test", "estuaire", "peste", "est", …]
```

2) ▪ Recherche des mots ayant le plus de trigrammes en commun avec le mot à corriger
   ▪ Troncature des résultats (distance de Jaccard)
   ▪ Tri des résultats (distance de Damerau-Levenshtein)

II. A) Méthode des trigrammes

- Temps d'exécution très réduit (< 0,5 s)

- Très efficace pour les longs mots
  MAIS peu performant pour les mots courts
  (2-6 lettres)

II. B) Méthode probabiliste

- Étude de la probabilité d'erreur sur un mot:
  - Soit m un mot, c est une correction si

$$c = \arg\max_{d \in D}\left(P(d|m)\right) = \arg\max_{d \in D}\left(\frac{P(m|d)P(d)}{P(m)}\right)$$
$$= \arg\max_{d \in D}(P(m|d)P(d))$$

Probabilité d'avoir voulu écrire *d* sachant qu'on a écrit *m*

Probabilité d'avoir écrit *m* sachant qu'on a voulu écrire *d*

II. B) Méthode probabiliste

$$c = \arg\max_{d \in D}\big(P(d|m)\big) = \arg\max_{d \in D}(P(m|d)P(d))$$

- $P(d/m)$ augmente avec $P(d)$
- Exemple: si m = "lvoir"

  $P(\text{"avoir"}) > P(\text{"lavoir"})$ ➔ "avoir" privilégié

II. B) Méthode probabiliste

1) Création d'un dictionnaire qui à un mot associe son nombre d'occurrences dans le corpus

II. B) Méthode probabiliste

1) Création d'un dictionnaire qui à un mot associe son nombre d'occurrences dans le corpus

2) ▪ Pour un mot m, on dresse l'ensemble des mots connus situés à une distance de m inférieure à 2
   - ▪ Tri des résultats en fonction de $P(d/m)$
   - ▪ Éventuelle troncature

II. B) Méthode probabiliste

- Rapidité d'exécution (< 1 s)

- Couvre une grande majorité des erreurs MAIS ne propose pas de correction pour les distances > 2

- Correction propre à un corpus donné

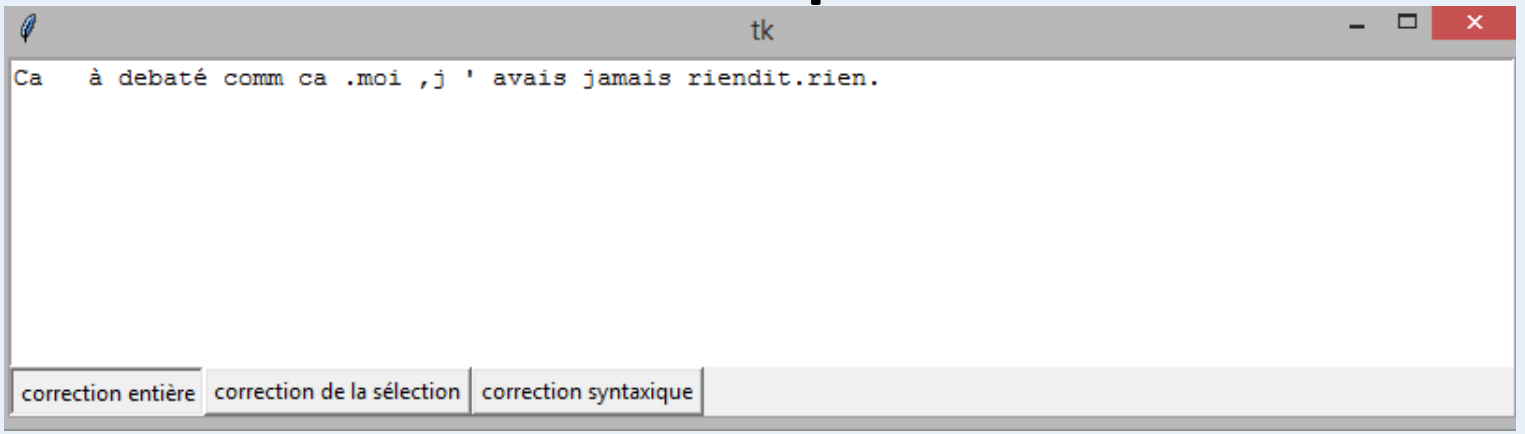
III. Algorithme final

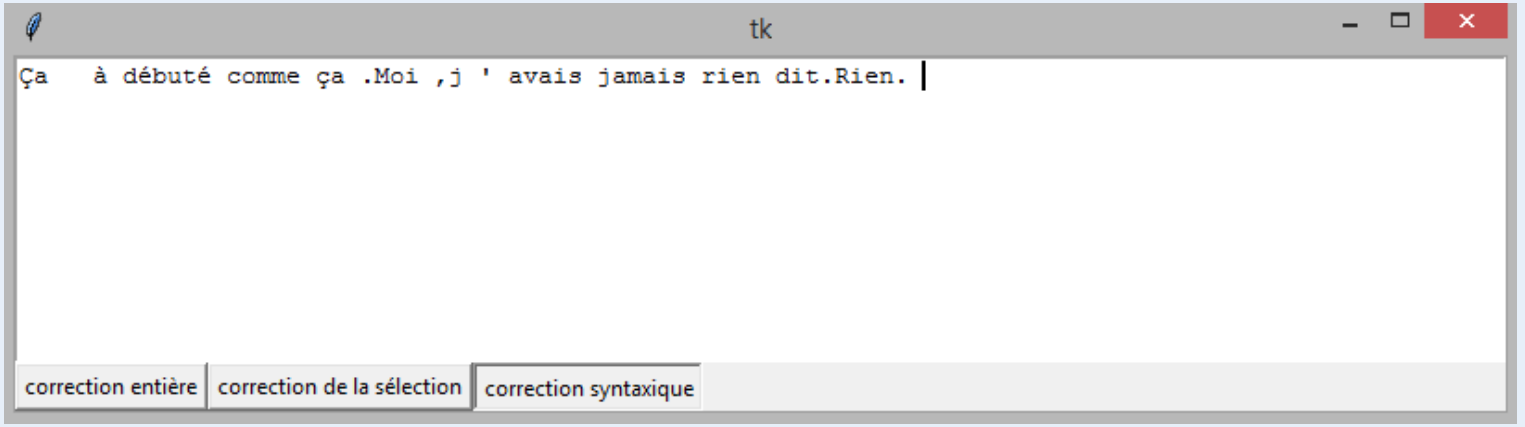
- Correction orthographique
  - Texte décomposé en liste de phrases
  - Chaque phrase analysée comme une suite de mots (corrigés un à un)
    - Longueur(mot) < 7 ⇒ méthode probabiliste
    - Longueur(mot) ≥ 7 ⇒ trigrammes

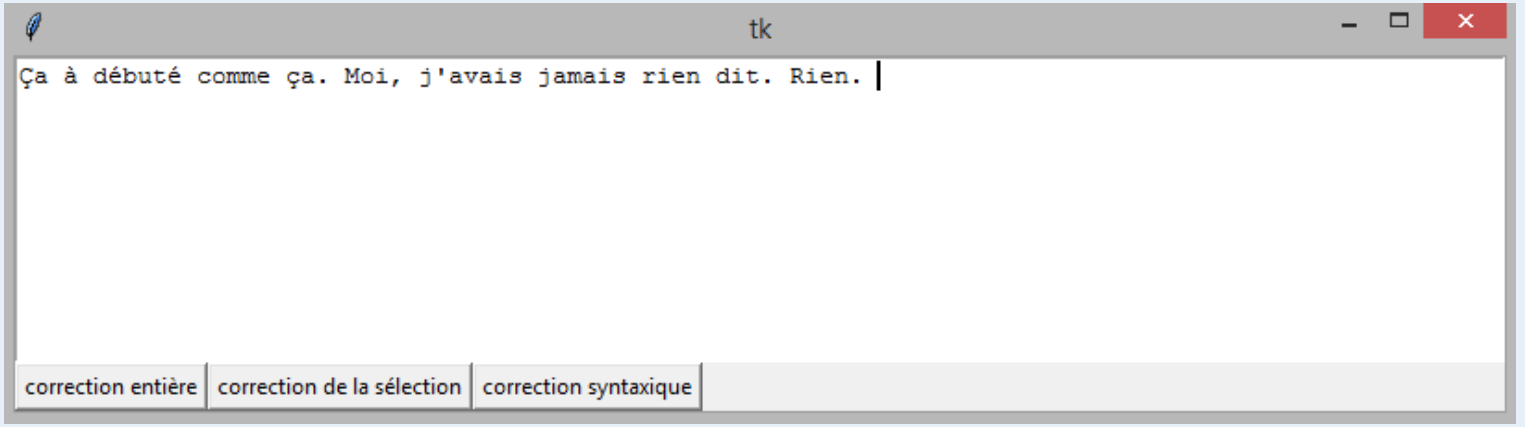
III. Algorithme final

- Correction syntaxique
  - Corrige la typographie (espaces, ponctuation, …)
  - Correction par pattern matching
  - Complexité en O(longueur du texte)

Exemple


tk
Ca à debaté comm ca .moi ,j ' avais jamais riendit.rien.
correction entière
correction de la sélection
correction syntaxique


tk
Ça à débuté comme ça .Moi ,j ' avais jamais rien dit.Rien.
correction entière
correction de la sélection
correction syntaxique


tk
Ça à débuté comme ça. Moi, j'avais jamais rien dit. Rien.
correction entière
correction de la sélection
correction syntaxique