Codage
Typographie

E. Jeandel

Emmanuel.Jeandel at lif.univ-mrs.fr

Typographie - Problèmes

- On essaie de convertir une suite de caractères en un texte imprimé.
- Comment afficher des mots ?
- Comment afficher les paragraphes ?

Aujourd'hui, uniquement le premier point.

Césures

But :

- Automatiser la division des mots
- Algorithme employé pour savoir où couper un mot à la fin de la ligne
- Algorithme employé pour savoir où il est possible de couper un mot

Règles usuelles

- Eviter de couper les mots (avec l'introduction de pénalités, par ex)
- Ne pas isoler une lettre en fin de ligne
- En renvoyer au moins trois à la ligne suivante
- Pas de césure en bas de page (pénalités aussi)
- Pas plus de 3 lignes consécutives avec un trait d'union (pénalités again).

Règles générales

- Couper entre deux con-son-nes sauf dans certains cas : nau-frage
- Cou-per entre une voyelle et une consonne simple
- Cas particuliers : gn,ch
- Avant une syllabe muette : ils galèrent
- Exception : Utiliser l'étymologie pour diviser (préfixes par ex)

Règles générales

- On arrive à donner un nombre petit de règles qui marchent dans 95% des cas
- Il en reste beaucoup
- Comment les traiter facilement ?

Première méthode

- Un dictionnaire qui contient tous les mots, et comment les couper.
- Avantage : fiabilité
- Avantage : sert de vérificateur orthographique
- Inconvénient : gros
- Inconvénient : rien de prévu pour les mots nouveaux

Première méthode

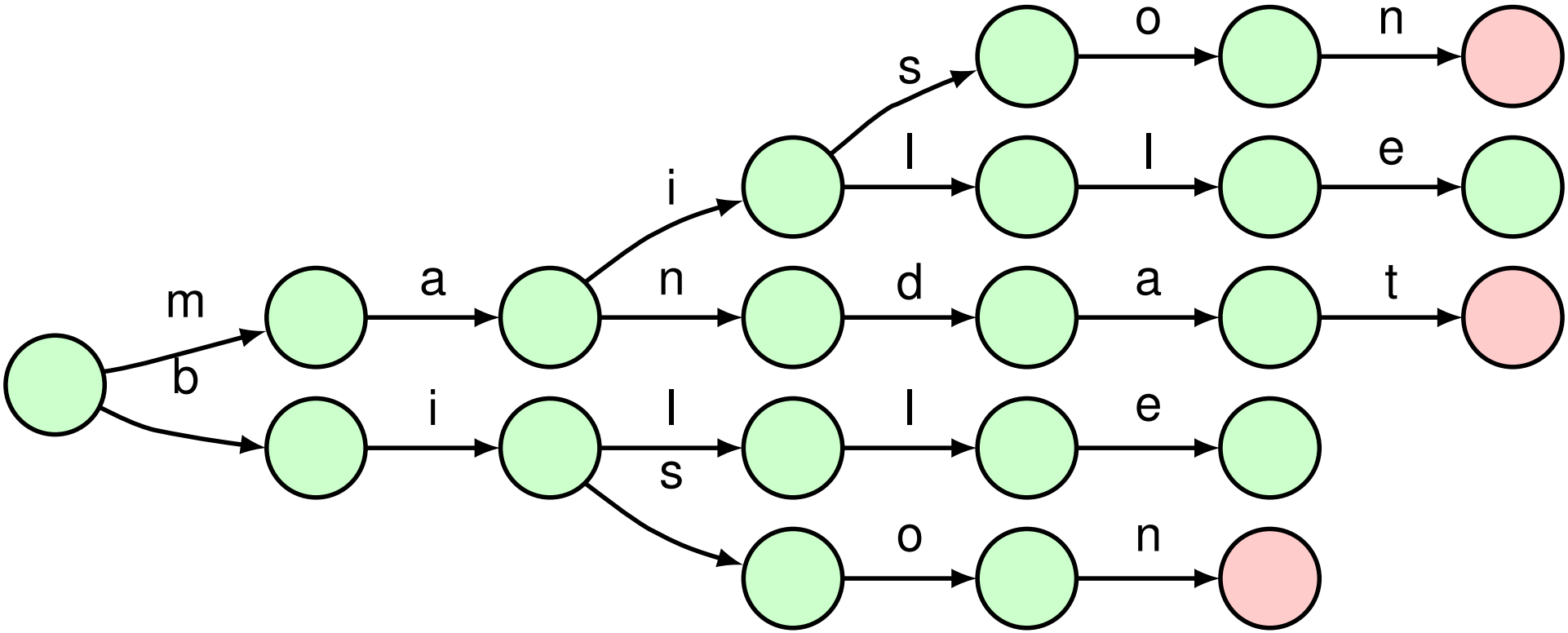
- Un dictionnaire qui contient tous les mots, et comment les couper.
- Avantage : fiabilité
- Avantage : sert de vérificateur orthographique
- Inconvénient : gros
- Inconvénient : rien de prévu pour les mots nouveaux

Structures de données

- Pour stocker les mots, il faut une structure de données efficace
- Trouver rapidement si un mot est présent dans la structure, et comment il faut le couper
- Structure de données petite
- Trie

Trie

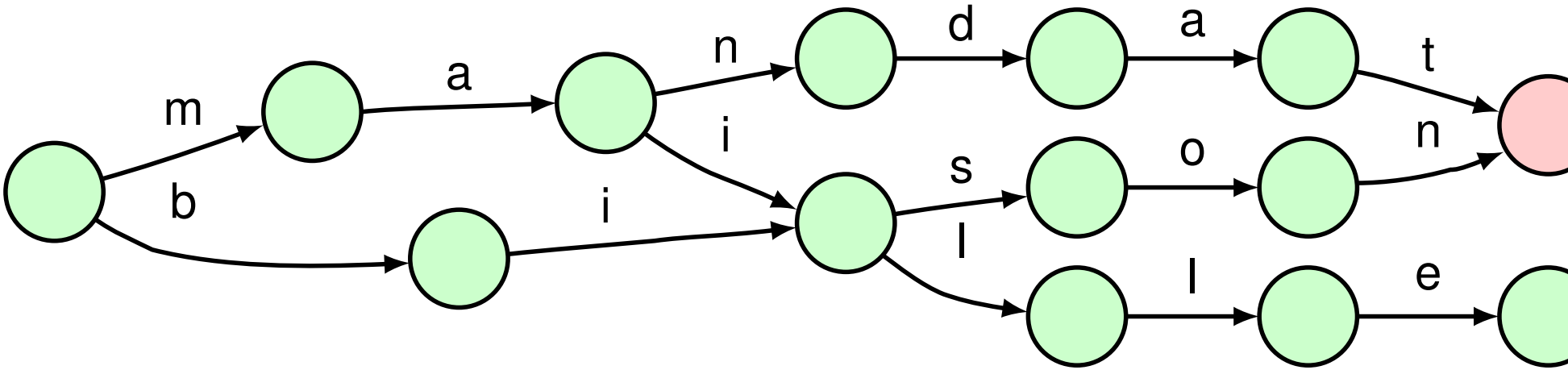
Pour les mots maison, bison, mandat, maille, bille



Structure utile aussi pour l'algorithme LZ78.

Optimisation

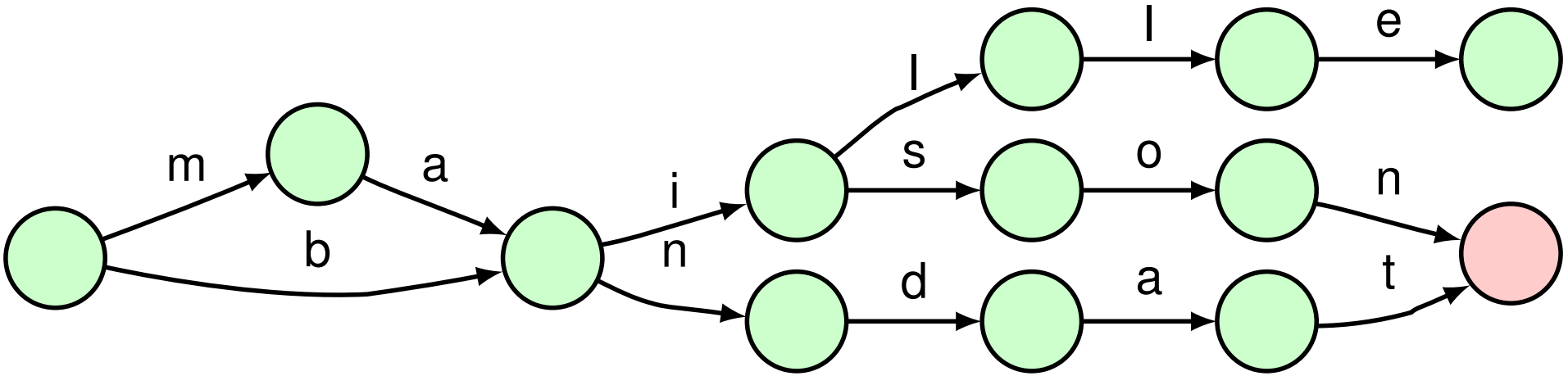
Pour les mots maison, bison, mandat, maille, bille.
Si jamais les mots se coupent de la même façon, on peut recoller certaines branches



Si jamais les mots se coupent de la même façon, on peut recoller certaines branches. (Sinon, il ne faut pas le faire)

Optimisation

Pour les mots maison, bison, mandat, maille, bille.
On peut optimiser un peu plus



En introduisant des nouveaux mots (ce n'est pas grave)

Deuxième méthode

- Un ensemble de cas où on peut couper un document
- Des cas particuliers pour les autres mots
- Avantage : Permet de faire quelque chose pour les mots nouveaux
- Inconvénient : S'il y a trop d'exceptions, ce n'est pas utile.

On peut voir la première méthode comme un cas particulier de la deuxième méthode.

Motifs

- Un motif est un mot coupé (par exemple) par un symbole spécial △

△hi

pupil△l

- Signification : dans tout mot qui contient ‘hi’, on peut couper avant le h
- Signification : dans tout mot qui contient ‘pupill’, on peut couper entre les deux l

Méthode

- Commencer par tous les motifs de taille 2
- Trois types de motifs : $\triangle$XY,X$\triangle$Y, XY$\triangle$
- Pour chacun des types de motifs, déterminer si c'est un "bon" motif.
- Un motif est un bon motif si le rapport

$$\frac{\text{mots qu'on peut couper à cet endroit}}{\text{mots qu'on ne doit pas couper à cet endroit}}$$

  est grand
- Si c'est un bon motif, on l'ajoute
- On passe aux motifs de taille 3, etc
- Problème de l'algorithme : il y a bien trop de motifs possibles de taille *n*.

Exemple

- Le motif $_{\triangle}$AB :

in-abor-dable	bio-dé-gra-dable
in-abor-dables	car-ros-sables
sur-abon-dance	char-geable
sur-abon-dant	coa-gu-lables
sur-abon-dante	com-muable
sur-abon-dantes	con-dam-nables
sur-abon-dants	con-so-lables
sur-abon-der	di-men-sion-nable
...	...
8 mots	1139 mots

$\Rightarrow$ On ne coupe donc pas.

Exemple

- Le motif $A_{\triangle}B$:

abo-mi-na-blement	bio-dé-gra-dable
abra-ca-da-brant	car-ros-sables
ca-ba-nons	char-geable
col-la-bo-ra-tion	coa-gu-lables
...	...
1699	1141

$\Rightarrow$ On ne coupe donc pas.

Exemple

- Le motif $N_{\triangle}N$:

aban-don-ner	aé-riennes
ap-pren-nent	ai-guillonnées
ban-ni-riez	cannes
brouillon-ner	con-sonnes
...	...
4557	926

$\Rightarrow$ On coupe donc.

Exemple

- Le motif $\triangle$ABO :

in-abor-dable	la-vabo
in-abor-dables	
sur-abon-dance	
sur-abon-dant	
sur-abon-dante	
sur-abon-dantes	
sur-abon-dants	
sur-abon-der	
8	1

$\Rightarrow$ On coupe donc.

Variante

- Commencer par tous les motifs de taille 2
- Pour chacun des mots, stocker quelque part tous les motifs de taille 2 qu'il contient
- Essayer ensuite chacun de ces motifs pour tester si c'est un bon motif
- Passer ensuite à la taille 3
- Pour la taille 3, il n'est pas nécessaire de stocker tous les motifs de taille 2 qui ont été sélectionnés à l'étape précédente

Algorithme

- Après l'algorithme, on obtient une liste de motifs qui expliquent où couper.
- Il y a (a priori) beaucoup de cas où ces motifs se trompent
- Dans un deuxième temps, faire exactement le même algorithme
- Parmi les mots où on se trompe, chercher des motifs pour lequels on se trompe beaucoup

Algorithme (suite)

- Exemple : Dans la première passe, on s'aperçoit qu'on peut couper en français avant *h* et *i* : $\triangle$ hi (prohiber, exhiber, vehicule)
- Mais on se trompe sur tous les mots en chi (enrichir, anarchie, machins)
- Dans la deuxième passe, ajouter le motif : c$\not\triangle$hi

Exemple

- Le motif $\not\triangle$NES :

éva-nes-cence	bagnes
éva-nes-cent	alignés
éva-nes-cente	ba-nanes
éva-nes-centes	ca-banes
...	...
41 mots	996 mots

$\Rightarrow$ On ne coupe donc pas.

Exemple

- Première étape :
  N$\triangle$N $\triangle$ABO ...
- Deuxième étape :
  $\not\triangle$NES ...

Les mots de la deuxième étape sont prioritaires sur les mots de la première étape.

Algorithme (fin)

- On a peut être, lors de la deuxième phase, interdit certaines coupures alors qu'on ne devrait pas
- Faire une troisième passe pour ajouter des coupures
- Faire une quatrième passe pour enlever des coupures
- Faire une cinquième passe pour ajouter des coupures
- etc. . .

Format

- Pour représenter les césures, on utilise le format suivant :

```
o1s2tas .en1o2
```

- Si on voit le motif “ostas”, alors il est permis (au niveau 1) de couper entre “o” et “stas” et interdit (au niveau 2) de couper entre “os” et “tas”
- Si on voit le motif “eno” (en début de mot), alors il est permis (au niveau 1) de couper entre “en” et “o” et interdit (au niveau 2) de couper après “eno”
- Pour trouver si on peut couper à un endroit donné, on teste tous les motifs, et on regarde le plus grand niveau qui explique ce qui se passe
- Si c’est un niveau pair (2,4...), on ne peut pas couper
- Sinon on peut couper

Exercice

- perfection
  `1pe .pe4r 1fe 1ti`
- village
  `il2l vil3l`
- recroqueviller
  `il2l vil3l uevil4l 1c2r 1ro 1q 4que. 1vi`

Exercice

- Empecher la coupure de mots à la syllabe “cons” au début du mot (comme dans construction) tout en permettant de couper en milieu de mot (dans reconstruction)
- res-treint , re-stru-cture, res-tau-rant, re-sur-gir, res-pect

Ce que le moteur de césure ne peut pas faire

- Trouver où couper un mot suivant son contexte
- président, ferment
- “ent” pose problème : quelquefois muet, quelquefois non

Typographie - Autres problèmes

- Dans des fontes italiques, certaines lettres en fin ou en début de ligne “dépassent” (p,f)
- On peut décider de changer très légèrement la taille de la fonte d’une ligne pour l’esthétique
- Comment en tenir compte dans l’algorithme total-fit ?

Typographie - Autres problèmes

- Il faut aussi réussir à transformer les mots en dessins
- Problème des ligatures (`ff`, `ffi` comme dans effacer, effiler)
- Les césures peuvent ajouter des ligatures et donc changer les colles, boîtes et pénalités (le mot raffiné)