

Normalisation & logiciel libre

Sébastien Paumier

`paumier@univ-mlv.fr`

Données linguistiques

- dictionnaires électroniques
- grammaires
- tables de lexique-grammaire

Dictionnaires électroniques DELAF

Mots simples:

point, .ADV+z1

poli, .A+z1:ms

poli, .N+z1:ms

poli, polir.V+z1:Kms

Mots composés:

cent soixante, .DET+Dnum:mp:fp

coup de sang, .N+NDN+z3:ms

coup de sifflet, .N+NDN:ms

coûte que coûte, .ADV+PCONJ+z1

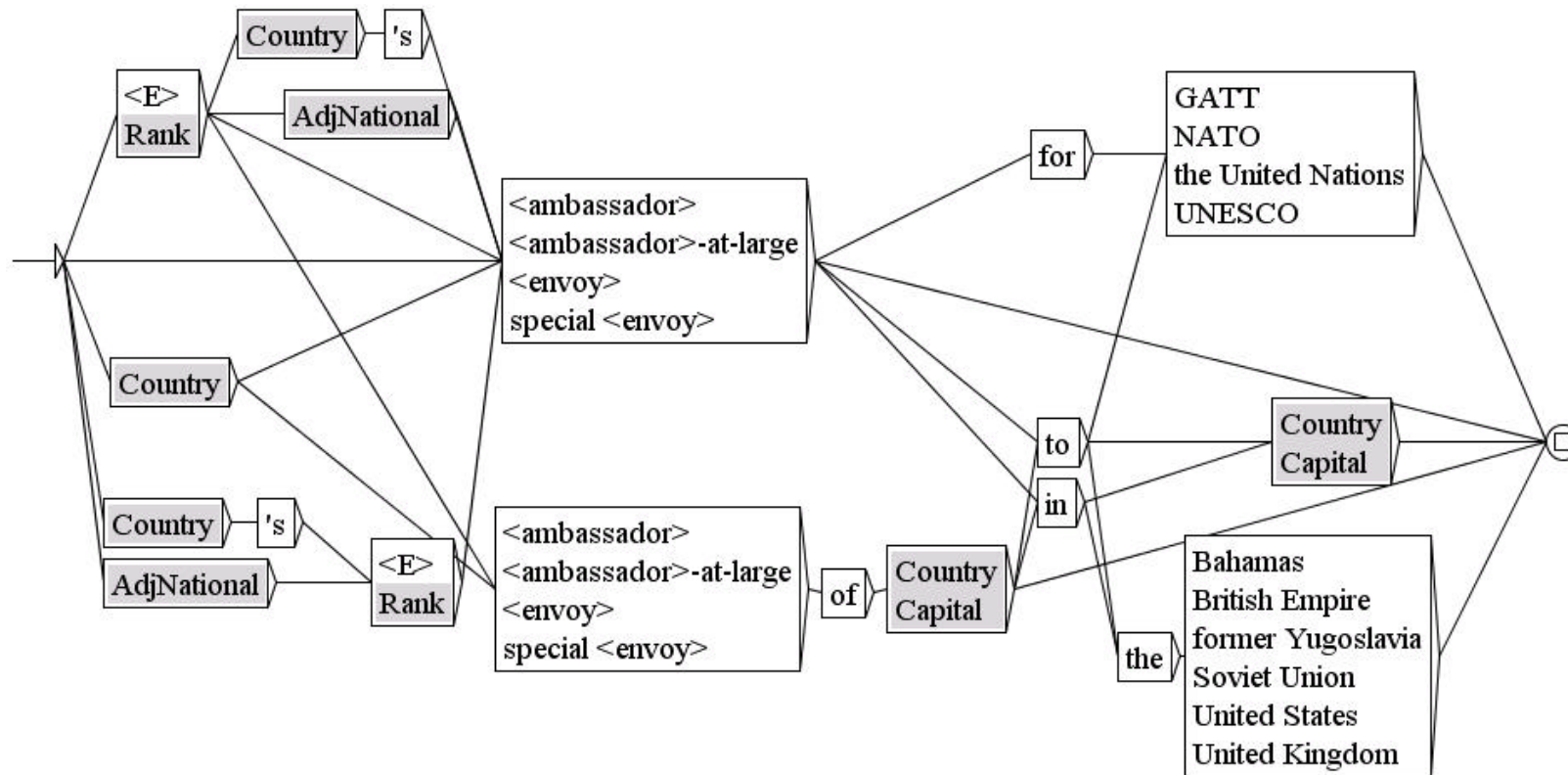
Dictionnaires électroniques DELAF

`équilibristes,équilibriste.N+z1:mp:fp`



- `équilibristes`: forme fléchie
- `équilibriste`: lemme
- `N`: code grammatical
- `z1`: code sémantique pour « mot courant »
- `mp`: code flexionnel pour « masculin pluriel »
- `fp`: code flexionnel pour « féminin pluriel »

Grammaires



Tables de lexique-grammaire

	N0 = Nhum	N0 = Nnr	N0 = V-n	N1 V	Aux = avoir	<ENT>	N1 = V-n	N0 V N1 de combien	N0 V N1 dans N2	N0 V Nhum sur ce point	N0 est Vpp W	N1 = Nconc	N1 = Nabst	<OPT>V-n (N0)	<OPT>V-n (N1)	<OPT>Exemple
+	+	-	-	+		abandonner	-	-	-	-	-	-	-	-	-	La chance a abandonné Max
-	+	-	-	+		abasourdir	-	-	-	-	+	-	-	-	-	Le bruit a abasourdi Max
+	-	-	-	+		abattre	-	-	-	-	-	-	-	-	-	La police a abattu le truand*
+	-	-	-	+		abattre	-	-	-	+	-	-	-	-	-	Le peuple a abattu le tyran*
+	-	-	-	+		aborder	-	-	-	-	-	-	-	-	-	Max a abordé une dame dans la rue
+	-	-	-	+		accolader	-	-	-	-	-	-	-	-	-	Le ministre a accoladé le héros
+	-	-	-	+		accompagner	-	-	-	+	-	-	-	-	-	Max a accompagné Léa
+	-	-	-	+		accoster	-	-	-	-	-	-	-	-	-	Max a accosté une dame dans la rue
+	-	-	-	+		accrocher	-	-	-	+	-	-	-	-	-	Jean a accroché une nana dans la rue*
+	-	-	-	+		accrocher	-	-	-	-	-	-	-	-	-	Les guerilleros ont accroché les soldats dans le défilé*
+	-	-	-	+		acheter	-	-	-	+	-	+	-	-	-	Max a acheté un député
+	-	+	-	+		administrer	+	-	-	+	-	+	-	administrateur	administré	Ce fonctionnaire a administré un grand nombre d'employés
+	-	-	-	+		adopter	-	-	-	+	+	+	-	-	-	Paul a adopté un petit Hindou
+	-	-	-	+		agrafer	-	-	-	-	-	-	-	-	-	Ce raseur a agrafé Max dans la rue
+	+	-	-	+		agresser	-	-	-	+	-	+	-	-	-	Max a agressé une passante
+	-	-	-	+		aimer	-	-	-	+	-	-	-	-	-	Max aime Ida
+	-	-	-	+		alarmer	-	-	+	+	-	-	-	-	-	Jo a alarmé les pompiers
+	+	-	-	+		aliter	-	-	-	+	-	-	-	-	-	Max a alité Luc* Loc N=lit]

Contrainte de temps en linguistique

- accumuler des données prend du temps
- les supports et les formats évoluent vite



Il faut raisonner à long terme sur une échelle
de plusieurs années

Maintenance à long terme

formats de données normés

+

logiciels libres

=

pérennité

Qu'est-ce qu'une norme ?

« Document établi par consensus et approuvé par un organisme de normalisation reconnu (ISO, CEI, UIT-T, ETSI, W3C, ...) »

Exemples: ANSI C, ASCII, XML

Qu'est-ce qu'un standard ?

« Norme de fait le plus souvent d'origine industrielle. Contrairement à une norme, un standard ne fait pas forcément l'objet d'une publication qui en détaille le contenu. »

Exemples: standard UNIX, standard EBCDIC, standard Unicode

Intérêts d'une normalisation

- description consultable et utilisable par n'importe qui
- existence d'une référence officielle faisant autorité
- favorise la portabilité des objets

Formats ouverts et fermés (1/2)

Spécification HTML

- tout le monde peut et pourra lire et écrire du HTML
- choix parmi de nombreux outils (navigateurs, éditeurs, ...)

Format .doc

- seul Word peut lire correctement un .doc
- peut-on toujours lire un fichier créé il y a 8 ans ?

Formats ouverts et fermés (2/2)

Quand les logiciels et les formats évoluent, il faut évoluer:

- coût financier dû à des incompatibilités matérielles et/ou logicielles
- nécessité de faire migrer les données, d'où perte de temps et risque d'erreurs

Maintenance des supports (1/2)

Quand les supports physiques changent, il faut aussi évoluer:

des données conçues pour un système à cartes perforées doivent être adaptées à un nouveau logiciel lorsqu'on veut les porter sur une autre machine

⇒ si le format est fermé, la portabilité est impossible à assurer

Maintenance des supports (2/2)

Les données linguistiques s'élaborent sur de longues périodes de temps

- En 30 ans: cartes perforées, bandes magnétiques, cassettes, disquettes 5^{1/4}, disquettes 3^{1/2}, CD-ROM, ZIP, DVD, ...
- L'utilisation d'un format ouvert normé permet d'assurer la pérennité des données

Non respect d'une norme (1/3)

Spécifications HTML

- applets, frames et formulaires qui ne marchent pas sur tous les navigateurs
- liens mal résolus
- tags exotiques (**blink**)

Non respect d'une norme (2/3)

Standard Unicode

	018	019	01A	01B	01C	01D	01E
0	Ḅ	Ɛ	Ɔ	ṁ	ǀ	ǃ	Ā
1	Ḃ	Ƒ	Ɔ	Ṁ	ǁ	ǃ	ā
2	Ḅ	ƒ	Ɔ	Ṁ	ǃ	ǃ	Ā
3	Ḅ	Ƒ	Ɔ	Ṁ	ǃ	ǃ	ā
4	Ḅ	Ƒ	Ɔ	Ṁ	ǃ	ǃ	ā
5	Ḅ	Ƒ	Ɔ	Ṁ	ǃ	ǃ	ā
6	Ḅ	Ƒ	Ɔ	Ṁ	ǃ	ǃ	ā

Dans la police Unicode par défaut de Windows (*Microsoft Lucida Sans Unicode*), le caractère **dž** n'est pas accentué et est affiché **dz**



impossible de lire correctement du serbo-croate sous Windows

Non respect d'une norme (3/3)

Majuscules accentuées en français

Officiellement: à Marne-la-Vallée ⇒ À MARNE-LA-VALLÉE

En pratique: cela varie selon les auteurs, les éditeurs, les types de textes, ...

Exemple: différences au sein d'un même article de journal (Libération du 17/11/2003)



eXtensible Markup Language

www.w3.org/XML/

- concept de *type de document*
- indépendant du matériel et des programmes
- ≠ HTML:
 - ensemble de tags non fixés
 - un document doit être validé par un parser
 - on ne s'intéresse qu'à la structure des données, pas à leur présentation

DTD: Document Type Declaration

```
<!ELEMENT examen (matière,exercice+)>  
<!ELEMENT matière (#PCDATA) >  
<!ELEMENT exercice (titre?,question+)>  
<!ELEMENT titre (#PCDATA) >  
<!ELEMENT question (#PCDATA) >
```

Exemple de document XML

```
<examen> <matière>Mathématiques</matière>
```

```
  <exercice><titre>Problème</titre>
```

```
    <question>
```

```
Démontrer que pour tout entier  $n > 2$ , on ne peut  
pas trouver  $a$ ,  $b$  et  $c$  entiers tels que  $a^n + b^n = c^n$ .
```

```
  </question>
```

```
</exercice>
```

```
<exercice><titre>Lire le sujet en entier</titre>
```

```
  <question>Ignorer l'exercice 1 et résoudre
```

```
l'équation  $X^2 - 4 = 0$ . </question></exercice>
```

```
</examen>
```

Text Encoding Initiative

www.tei-c.org

DTD XML décrivant une façon de coder les textes

Exemple d'architecture de document:

```
<TEI.2>
  <teiHeader> [ TEI Header information ] </teiHeader>
  <text>
    <front> [ front matter ... ] </front>
    <body> [ body of text ... ] </body>
    <back> [ back matter ... ] </back>
  </text>
</TEI.2>
```

Phénomènes pris en compte

- type de texte: roman, poème, article, ...
- hiérarchie: pages, chapitres, strophes, ...
- mots étrangers, citations, ...
- notes, renvois, références, ...
- NdT, notes de corrections typographiques
- abréviations, nombres, dates, ...
- etc.

Le standard Unicode

www.unicode.org

Unicode associe un numéro unique à chaque caractère:

- quelle que soit la machine
- quel que soit le système d'exploitation
- quel que soit le programme utilisé
- quelle que soit la langue

Encodage des caractères

- UTF-8 (longueur variable)
- UTF-16 Little-Endian (2 octets)
- UTF-16 Big-Endian (2 octets)

- UTF-16 limite le nombre des caractères à 65536, mais cela suffit à représenter toutes les langues y compris le chinois et le coréen

Codage UTF-8

- pas d'en-tête
- pas de restriction sur le nombre de caractères
- ordre de lecture/écriture des octets indépendant de la machine
- accès séquentiel au n^e caractère

Codage UTF-16

- en-tête déterminant l'ordre des octets dans le fichier: `FEFF` (BE) ou `FFFE` (LE)
- caractères de `0000` à `FFFF` codés sur 2 octets, d'où accès aléatoire au n^e caractère
- problème: l'ordre de lecture/écriture des octets dans les primitives d'E/S varie selon les machines (PC: LE, Mac: BE)

Problèmes malgré Unicode

- pas de convention sur les retours à la ligne:
Windows `\r\n`, Linux `\n`, MacOS `\r`
- en UTF-16, l'ordre de lecture/écriture des octets doit être bien géré selon le système et/ou la machine (Word ne sait pas lire un fichier Big-Endian)

Solution

- utiliser les bibliothèques existantes lorsqu'elles vérifient exactement une norme commune (contre-exemples en C: '\n', <wchar.h>)
- écrire son propre code portable sinon (exemples: JVM, Unitex)

Dépendances données/programmes

- format fermé \Rightarrow programme propriétaire
- format ouvert:
 - au pire, on peut toujours réécrire un logiciel pour relire le format
 - au mieux, le code-source est disponible, ce qui évite de réinventer la roue

Principe des logiciels libres

- liberté d'utilisation pour n'importe qui et pour n'importe quel usage
- liberté d'étudier et d'adapter le logiciel
- liberté de copie et de distribution
- liberté d'améliorer le logiciel et de diffuser ces améliorations

www.gnu.org

Avantages des logiciels libres

- l'accès au code source garantit la maintenabilité et la portabilité des programmes
- souvent plus fiables que les logiciels propriétaires
- souvent gratuits

Exemples de logiciels libres

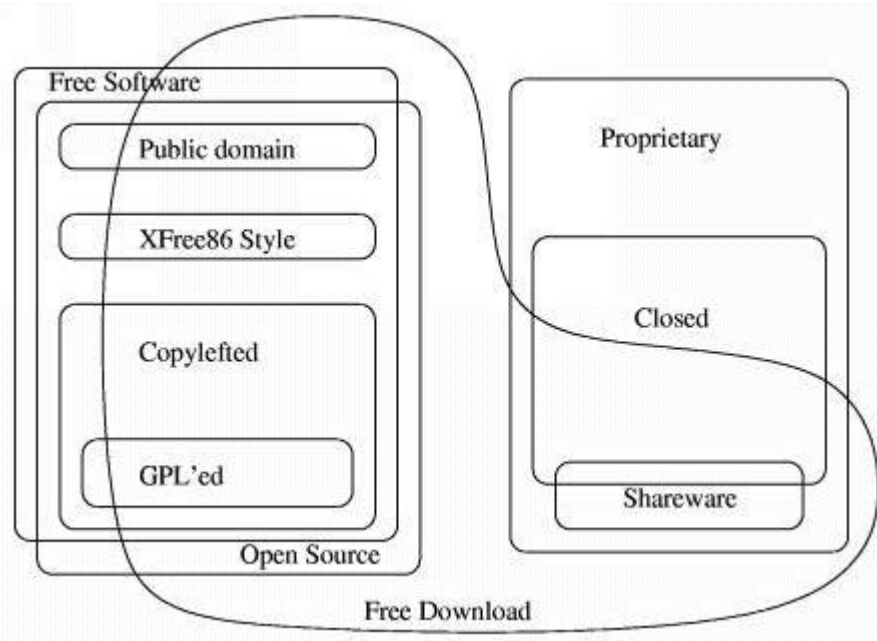
- GNU Linux
- g++
- The GIMP
- KOffice
- Gnome
- Mozilla

Free Software/Open Source

Même approche pratique, philosophies différentes:

« Open source is a development methodology; free software is a social movement. »

Free Software/Open Source



Les différentes catégories de logiciels

(<http://www.gnu.org/philosophy/categories.html>)

Le Copyleft

- garantir les libertés de l'utilisateur
- préserver l'accès à l'œuvre
- ≠ domaine public:
 - on peut faire ce qu'on veut d'une chose qui est dans le domaine public
 - une chose « copyleftée » doit le rester

Licences compatibles GPL

- GNU GPL: General Public License
- GNU LGPL: Lesser General Public License
- X11 license
- W3C Software Notice and License
- Berkeley Database License

Licences Free Software non-compatibles GPL

- OpenSSL license
- Open Software License, version 1.0
- Apache License, Version 1.1
- Common Public License Version 1.0
- Mozilla Public License (MPL)
- Sun Public License

Licences non Free Software

- Sun Community Source License
- Open Public License
- University of Utah Public License
- Scilab license
- AT&T Public License

Besoins actuels

- normes XML pour représenter les données:
 - dictionnaires électroniques
 - grammaires
 - tables de lexique-grammaires
- outils pour manipuler ces données