Données linguistiques et corpus

Cours III

Grégoire Winterstein
Laboratoire Structures Formelles du Langage, Université Paris 8

gregoire.winterstein@linguist.jussieu.fr

Université Paris Sorbonne

Détails pratiques

Cours

- Le Jeudi, **10h30-12h30**, salle 220 (Serpente)
- **Séances prévues** :

~~16 février~~	~~23 février~~	**8 mars**	15 mars
22 mars	29 mars	5 avril **??**	12 avril

- Site web :
  `http://gregoire.winterstein.free.fr/Ens/DonneesLing/`

Évaluation

- Un examen sur table (séance à préciser).

Thèmes abordés dans le cours

- Quelles données pour la linguistique ? ✓
- La recherche par expressions régulières. ✓
- **Les formats d'encodage des textes ?**
- Étude d'outils particuliers :
  - **TLFi**
  - **Frantext**
  - French Treebank
- L'approche expérimentale :
  - Mise en place d'une expérience.
  - Discussion des résultats.

Projet d'expérience

En prévision de la mise en place d'une expérience :

- Réfléchir à des phénomènes linguistiques que vous désirez tester.
- Construire des paires minimales mettant en jeu les phénomènes en question.

Contenu

Recherche avancée

2 onglets (haut de la page)

Recherche assistée : 5 façons de définir une recherche.

- Spécialisation par discipline, code grammatical, emploi (invariable, argotique. . .)
- Possibilité d'utiliser des **codes de contenu** dans le cadre 1 (cf. infra)
- Contraintes liées aux éléments de l'entrée (définition, langue d'emprunt. . .)

Recherche complexe : jusqu'à 6 contraintes liées sur une même recherche

- **Exemple** : verbes à usage péjoratif empruntés à l'italien et y repérer la définition appropriée.

Listes de mots

- Utile pour créer des listes de mots possédant des propriétés similaires.
- Ces listes sont ensuite réutilisables dans les recherches complexes (cf. code `&l`).
- À la création d'une liste, son contenu s'affiche dans le cadre 1.
- Utilisation des **expressions régulières** (cadre 4) :
  - Caractère joker : .
  - Répétition : $*$, $+$, ? et \1 (p.ex. `(co)\1` renvoie *coco*)
  - Disjonction : |
  - Ensemble de caractères : []
  - Négation : [$^{\wedge}$*xyz*]

Codes de contenu

- `&cxxx` : toutes les formes de l'infinitif *xxx*
- `&mxxx` : toutes les formes de l'adjectif ou substantif *xxx*
- `&lxxx` : tous les membres de la liste nommée *xxx* (cf. infra)
- `&q` : n'importe quel mot
- `^xxx` : toute séquence qui ne correspond pas à la séquence *xxx*
- `&nxxx` : toute séquence qui ne contient pas *xxx*
- `&dn xxx` : signifie que le contenu *xxx* doit se trouver à *n* mots au plus en partant du début de l'objet étudié.
- `&fn xxx` : signifie que le contenu *xxx* doit se trouver à *n* mots au plus en partant de la fin de l'objet étudié.
- `|` relie des contraintes multiples (**attention** $\neq$ regexp)

Où utiliser les codes de contenu ?

Recherche assistée :

- Cadre 1 : cherche uniquement dans les vedettes
- Cadre 5 : cherche dans le contenu de l'objet spécifié

Recherche complexe Cadre "contenu" : cherche dans le contenu de l'objet spécifié (il n'est pas toujours pertinent d'utiliser un code de contenu selon le type de l'objet).

- `&cmanger`
  - En recherche assistée
  - En recherche complexe
- `&cmanger &q &mpain`
- `^&cruer dans les brancards`

Exercices

1. trouver tous les verbes empruntés à l'arabe
2. trouver tous les noms qui riment avec *rieur*
3. trouver tous les mots en *ing* empruntés à l'anglais
4. trouver toutes les expressions figées basées sur le verbe *attraper*, du type *attraper froid*
5. trouver tous les mots échos (p.ex. *bonbon*, *chouchou*...)
6. trouver tous les palindromes qui contiennent entre 6 et 8 lettres

Formats d'encodage

Coder de l'information

- **Code binaire** un **bit** a deux valeurs : 0 et 1.
- Un **octet** : une suite de huit bits.
- Un octet permet de coder $2^8 = 256$ possibilités
- Deux octets permettent de coder $2^{16} = 65536$ possibilités

Coder de l'information textuelle

- On a recours à des **jeux de caractères**, définis par des normes :
  - Un jeu = un ensemble de caractères, identifiés par un numéro.
  - Au niveau du codage, un texte est une suite de numéros indiquant le caractère à représenter.
  - Pour afficher/traiter correctement un texte, il est nécessaire de connaître le jeu de caractères correspondant à l'encodage.

Différents jeux de caractères

- `ASCII` : la table basique qui contient le nécessaire pour transcrire l'anglais. 127 caractères en tout = codage sur **un unique octet**.
- L'`ASCII` sert de base à de nombreux jeux plus étendus, toujours codés sur **un unique octet**.
  - `Latin-1` (ou `ISO-8859-1`) : jeu de caractères adapté aux langues d'Europe de l'Ouest (contient les caractères accentués nécessaires au français), standard sur le web.
  - `Windows-1252` : jeu utilisé par Microsoft pour ses systèmes d'exploitation en Europe de l'ouest (proche du `latin-1`).
  - `MacRoman` : jeu (anciennement) utilisé par Apple pour ses systèmes d'exploitation en Europe de l'ouest.

  $\Rightarrow$ des jeux de caractères différents contiennent les mêmes caractères, mais pas aux mêmes index...
- `Unicode/UCS` : table "universelle" de caractères abstraite (= **norme**), vocation à pouvoir indexer tous les caractères possibles.
  - Peut-être encodé de différentes façons. Les plus courantes étant `UTF-8` et `UTF-16`.
  - Codage sur un nombre variable d'octets : entre **un et quatre**.

J'ai un problème, ça s'affiche mal

Votre texte est mal décodé. Que faire ?

1. Chercher d'où provient le texte source et son encodage probable.
   - Windows : fortes chances pour le `windows1252/latin-1`
   - MacOs : UTF-8 ou MacRoman selon l'âge de la machine sur laquelle le texte a été produit.
   - Systèmes Unix : très probablement de l'UTF-8
   - Si c'est un texte de corpus, de la documentation spécifiant l'encodage est généralement fournie.
   - Un document écrit au format `xml` spécifie normalement son encodage.
2. Examiner les caractères qui s'affichent mal :
   - « `Je chantÈ Áa dans ma tÍte` », même nombre de caractères mais pas les bons ⇒ mauvais jeu de caractère mais encodage sur le même nombre d'octets (p.ex. vous lisez du `latin1` comme du `macroman`).
   - « `Je chantÃ© Ã§a dans ma tÃªte` », plus de caractères que voulu ⇒ UTF-8 interprété comme un jeu de caractère à octet unique.
   - « `Je chant□□ a dans ma t□¬` », moins de caractères que voulu ⇒ jeu à octet unique interprété comme de l'UTF.

Changer l'encodage

Une fois que vous avez une bonne idée du jeu de caractères utilisé à la base pour votre texte, comment l'afficher correctement ?

- Dans un bon éditeur de texte (emacs, NotePad++...) la conversion peut se faire directement dans l'éditeur s'il n'a pas déjà correctement deviné le format d'encodage.
- Systèmes Unix (Linux, MacOs...) : utiliser la commande `iconv` (dans le Terminal de commance)
- Quand vous produisez des textes (e.g. e-mail) essayez de configurer votre outil pour qu'il fournisse le format désiré.

Afficher proprement un texte

- À supposer que le texte soit correctement décodé, reste la question de la police.
- Même en connaissant le caractère à afficher, il se peut qu'il ne soit pas disponible dans les glyphes disponibles au sein de la police qu'on utilise.
  - Certaines polices "fantaisistes" n'incluent pas les caractères accentués et se limitent au jeu standard de l'ASCII.
  - Il est rare que tous les glyphes UTF-8 soient disponibles dans une police donnée.
  - Comportement de certains traitements de texte : remplacer les caractères manquants par leur version dans la police d'affichage par défaut.
- Test sur la page d'accueil de Wikipedia.

Frantext (www.frantext.fr)

- Base de données de textes, majoritairement littéraires et philosophiques, en français.
- 4084 textes en tout.
- Période : de 1180 à 2009 (820 textes postérieurs à 1950).
- Deux versions principales :
  1. Frantext général : totalité des textes.
  2. Frantext catégorisé : 1200 textes **étiquetés** en partie du discours.
- L'accès à Frantext est **payant**. Pour y accéder :
  - Depuis les locaux de l'université.
  - Via les services à distance de la bibliothèque de Paris 4.
    `http://www.paris-sorbonne.fr/les-bibliotheques/nous-vous-proposons/`
    Voir 'Accès à distance' et suivre les instructions.
  - Via les services de la BIU : `https://www.biu.sorbonne.fr/biu/`

Consulter Frantext

1. Définir son corpus travail
   - Choix des textes au sein desquels faire ses recherches.
   - Par défaut, aucun texte n'est sélectionné.
   - Possibilité de sauvegarder son corpus (la composition, pas le contenu) via la fonction `Exporter`
2. Rechercher dans les textes : 4 possibilités
   - Texte exact
   - Flexion d'un verbe, d'un substantif ou d'un adjectif (id. TLFi)
   - **Expression de séquence**

Expressions de séquence

Suite de **descriptions** de mots en partie inspirée de la syntaxe des expressions régulières et réminiscente des codes du TLFi :

- Graphie exacte d'un mot, p.ex. `mangeront`
- Disjonction `(X|Y)` : tout élément qui vérifie la description X ou Y
- `&?xxx` : un élément optionnel qui vérifie la description *xxx*
- `&cxxx` : toutes les formes de l'infinitif *xxx*
- `&mxxx` : toutes les formes de l'adjectif ou substantif *xxx*
- `&lxxx` : tous les membres de la liste nommée *xxx* (créée comme dans le TLFi, cf. barre de gauche)
- `&q(`n_1`,`n_2`)` : une suite de mots quelconque dont le nombre d'éléments est compris entre n_1 et n_2 (raccourci : `&q=&q(1,1)`)
- `^xxx` : tout élément qui ne correspond pas à la description *xxx*
- `&e(xxx)` : tout élément qui vérifie les informations de catégorisation spécifiée par *xxx* (uniquement pour la recherche dans Frantext catégorisé), cf. infra

Informations de catégorisation

Utilisation de la description `&e()` :

- `&e(g=X Y)` : tout élément qui possède la catégorie *X* ou *Y*
- `&e(c=X)` : tout élément catégorisé qui vérifie l'expression de séquence *X*
- `&e(g!=X Y)` : tout élément qui ne possède ni la catégorie *X* ni la catégorie *Y* (négation identique pour `&e(c!=X)`).
- Combinaisons possibles :
  `&e(c=(&mdemi &q | &q &mdemi) c!=&mheure g=S)`

Codes grammaticaux courants :

`A`	Adjectif	`Inf`	Infinitif
`Adv`	Adverbe	`P`	Pronom
`Cc`	Conj. coordination	`Pp`	Préposition
`D`	Déterminant	`S`	Substantif
		`V`	Verbe

Voir la liste complète dans l'aide de Frantext.

Exercices (I)

1. Quelle est la première attestation de **morphème** dans le corpus ?
2. Georges Perec utilise-t-il le terme **ordinateur** ?
3. Au présent de l'indicatif, quelle est la forme la plus fréquente du verbe **subir** ?
4. Déterminez si l'inversion du sujet est possible après **aussitôt que**.
5. Peut-on insérer un élément entre **tandis** et **que** dans la « conjonction de subordination » **tandis que** ?
6. Trouvez tous les exemples dus à Balzac d'utilisation d'un substantif qui est également une forme du verbe **voir**.

Exercices (II)

1. L'expression **faire amende honorable** s'emploie-t-elle au passif ?
2. La forme **sue** est-elle plus souvent une forme du verbe **savoir** ou du verbe **suer** ?
3. De quand date le substantif **déterminant** pris dans son sens linguistique ?
4. A quelle fréquence rencontre-t-on la négation exprimée par **ne** sans **pas** (par rapport à la négation avec **pas**) pendant la première moitié du XX^{e} siècle ?
5. Comparez la fréquence des verbes du premier groupe, à la première et à la troisième personnes du pluriel, au passé simple, au XIX^{e} et au XX^{e} siècles.