Construire un corpus structuré

- Noms de fichiers
- Codes caractères
- Nettoyage
- Metadata
- Prétraitements
- Les choix OMNIA
- Insertion dans une base de données
- Architecture générale

Noms de fichiers

- Élément de base ! À mettre en place correctement dès que possible.
- Respecter des **règles d'écriture très strictes** pour éviter tout problème technique ultérieur :
  * uniquement minuscules (bdc) et nombres (digits), aucun caractère accentué, aucun caractère diacritique, jamais de blanc (souligné)
  * l'**ordre alphabétique des noms de fichiers** sera présent partout : si l'**ordre chronologique** importe, placer en tête du nom une indication de date (et vérifier que le codage fonctionne)
  * un codage interne du nom (auteur, titre, catégorie quelconque) permettra de gagner beaucoup de temps à toutes occasions, trouver un codage simple et compréhensible.
- Toute erreur à ce stade se paye cher, et l'on éprouve ensuite les pires difficultés à remettre de l'ordre.

Codes caractères

- En informatique, les caractères alphanumériques sont codés en codage binaire ; il existe une grande quantité de codages différents.
- Les plus courants : ASCII, windows-1252, iso 8859-15 (latin-15), utf-8.
- Le seul qui soit à peu près universel, le plus efficace et le plus solide est l'**utf8** (Universal Character Set Transformation Format - 8 bits). Tous les grands logiciels le reconnaissent, certains l'exigent. **Il faut impérativement coder en utf-8**, pas forcément facile.
- Il faut autant que possible vérifier l'encodage dans le fichier récupéré, et convertir si nécessaire (commande iconv).

Nettoyage

- Les fichiers dont on dispose au départ peuvent être dans les formats les plus variés : texte pur, html, odt, doc, rtf, pdf…
- Il faut **tout convertir en format texte pur** pour pouvoir les manipuler avec un « éditeur » ordinaire (geany, gedit).
- Cela fait, un examen très attentif de la structure est indispensable. Il s'agit de savoir comment sont construits et constitués les fichiers dont on dispose : éléments inutiles, codes bizarres (type html), balises plus ou moins utiles ; il faut identifier les éléments caractéristiques qui peuvent permettre les opérations de nettoyage et de mise en forme minimale par procédure. Dans de nombreux cas, les fichiers comportent d'origine des informations annexes utiles, qu'il faudra récupérer d'une autre manière que le texte lui-même.
- Si l'on n'a que quelques dizaines de fichiers, un nettoyage manuel est envisageable. Mais, même dans ce cas, **un nettoyage par procédure est plus sûr et plus homogène**.
- Il faut alors consacrer le temps nécessaire (!) à l'écriture d'un script ad hoc. Pas de nettoyage sérieux sans un minimum de programmation.

Metadata

- Toute base de données textuelles (corpus) inclut des « **informations bibliographiques** » sur les textes qu'elle contient, on utilise couramment le terme metadata.
- Les catégories habituelles sont connues de tous : auteur, titre, date.
- Mais un corpus bien structuré comporte presque nécessairement de nombreuses autres informations, qui seront décisives au moment de l'emploi et de l'analyse du corpus : genre, zone géographique, prose/vers…
- Là encore, il vaut mieux **mettre en place ces metadata en début de parcours**, toute réfection demandera plus d'efforts.
- En général, **il vaut mieux trop d'informations que pas assez**. Combiner ou regrouper des informations disponibles est beaucoup plus simple que d'opérer des distinctions à l'intérieur de catégories existantes.
- On doit cependant souligner qu'il existe des méthodes de text mining (recherche de mots-clés, clustering) qui sont précisément destinées à identifier des catégories, on pourra le plus souvent les intégrer, quoique pas toujours simplement.

Metadata 2

- Pour les données historiques, le **codage des dates** est des plus importants ; bien entendu, si l'on connaît l'année et que l'on peut s'en contenter, la situation n'est pas trop complexe. Mais si les dates sont des fourchettes, énoncées selon des procédures différentes (1085-1095 vs 1e moitié 12e), cela se complique singulièrement ; **il n'existe pas de solution générale connue**, il faut procéder au coup par coup. Le codage des éléments non-datés est particulièrement délicat.
- Beaucoup de logiciels attendent **les metadata dans un fichier séparé**. Même si ce n'est pas le cas, il est de beaucoup préférable d'avoir un fichier structuré regroupant ces informations (format csv en général).
- Si les fichiers sont nombreux (plus de 500), il est utile de disposer de ces metadata dans **une petite base de données séparée**, que l'on puisse interroger à l'aide de requêtes complexes. Le format sqlite est très performant, il existe un logiciel d'interface puissant et facile à utiliser, **sqlitestudio**. (http://sqlitestudio.pl/)
- Principe fondamental : **pour les textes anciens, il n'existe pas, et il ne peut pas exister de codage universel**, chaque base de données exige un codage spécifique.

Prétraitements

- Dès lors que l'on dispose de fichiers textes nettoyés en utf8, on procède au prétraitement, qui comporte **deux phases essentielles** : la tokenisation et le pos-tagging (ou lemmatisation).
- **Tokeniser** consiste à découper le texte en unités élémentaires, ce qui est bien plus complexe qu'un découpage « par mots ». Le nombre de questions précises à définir est très élevé. Les choix effectués ici conditionnent la suite.
- Lorsque le fichier est réduit à une suite de tokens séparés (en général un token par ligne) on procède à l'**enrichissement de chaque token** : on détermine son POS (part-of-speech = catégorie grammaticale) et son lemme, plus éventuellement quelques autres caractères.

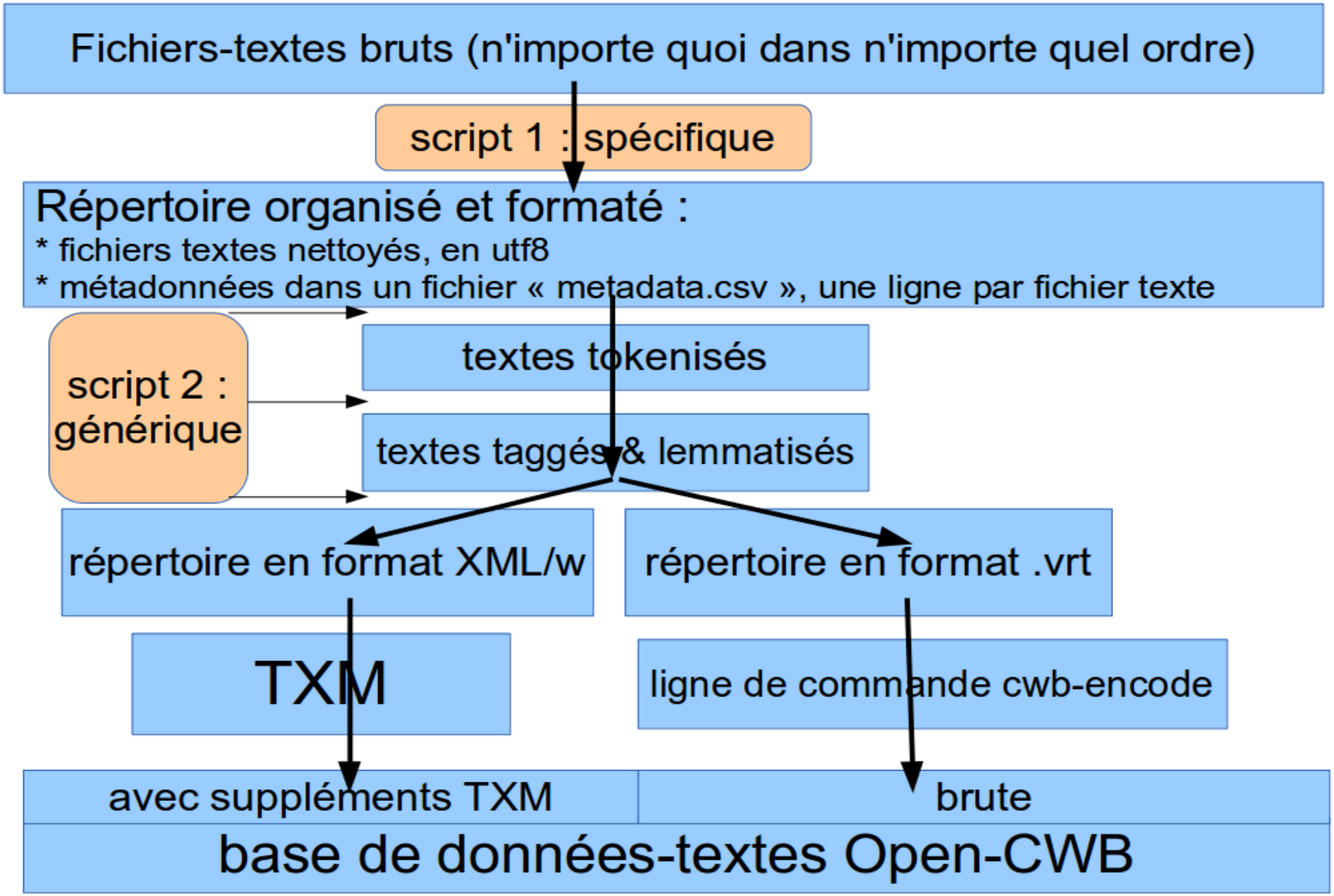
Les choix OMNIA

- Le latin, et en particulier le latin médiéval, posent, pour les prétraitements, de redoutables problèmes : toute une série de choix délicats sont inévitables.
- Non seulement la **morphologie latine est très riche**, mais les textes se présentent avec une **incroyable variété de graphies**, qu'un traitement informatique ne peut pas accepter utilement.
- Il faut avant tout adopter des règles de graphie précises, **un stock de lemmes déterminé**, et **une liste de POS** claire. Les choix des latinistes regroupé sous l'appellation OMNIA ont privilégié la simplicité : réduction des différences i-j et u-v, réduction des diphtongues, 12 pos et pas davantage. Le choix principal a été de s'en tenir pour l'essentiel aux entrées du dictionnaire réalisé sous la direction de Michel PARISSE, *Lexique Latin-Français Antiquité et Moyen-Age*, Paris, 2006 (environ 57000 lemmes).
- Ces choix ont été incorporés dans un « fichier de paramètres » qui permet à un logiciel précis, **tree-tagger**, de lemmatiser les textes latins. La listes des formes incluses dans ce fichier est énorme (plus de 3,2 millions) et pourtant cette liste est très loin d'être complète. Tous les outils nécessaires sont en libre accès et en open-source (www.glossaria.eu). Le taux d'erreur de la lemmatisation automatique est faible (de l'ordre de 3%) mais pas négligeable (tree-tagger tient compte principalement de l'ordre des mots, ce qui ne va pas de soi pour des textes latins). De toutes manières, ce fichier de paramètres peut et doit encore être amélioré.

Insertion dans une base de données

- Les fichiers nettoyés, prétraités et munis de leurs metadata doivent enfin être inclus dans une base de données proprement dite.
- Il existe une grande variété de « moteurs d'indexation », aux propriétés différentes. Tout le monde connaît et emploie les moteurs SQL généralistes, implémentés dans MySQL et Postgre-SQL. On préconise ici l'emploi d'**open-CWB** (open corpus workbench), spécialement destiné à gérer de très grandes bases de données textuelles complexes. On peut employer ce logiciel soit **en ligne de commande** (rapide et efficace), soit au travers d'interfaces graphiques dédiées, **TXM** (présenté ici) ou **CQP-Web**. On peut aussi employer les bases cwb directement à partir de logiciels statistiques, comme R.
- Ces logiciels reconnaissent chacun un ou plusieurs formats spécifiques. Il faut donc procéder à une dernière étape préalable, la mise en forme. CWB et TXM attendent des formats différents (au moins jusqu'ici)…
- L'ensemble des prétraitements n'est pas envisageable manuellement. Il faut agir par procédure. TXM a été conçu pour exécuter lui-même l'essentiel de ces prétraitements (quoiqu'il accepte des fichiers prétraités), mais la mise en place des paramètres nécessaires n'est pas immédiate. Dans l'ensemble, il paraît raisonnable de procéder par programmes du début à la fin. On écrira un ou plusieurs scripts destinés au nettoyage et à l'encodage ; là, tout dépend de la nature et de l'organisation des fichiers récupérés. Dans un second temps, un autre script, qui peut avoir un emploi très général, se chargera de la tokenisation, du pos-tagging et de la mise en forme finale.

Architecture générale



Conclusions

- Constituer une base de données de textes anciens est une opération longue.
- La fabrication d'un corpus structuré nécessite à la fois beaucoup de mécanique et beaucoup de réflexion.
- Il faut impérativement respecter quelques règles précises
- De nombreuses difficultés ne sont pas résolues, c'est un champ de recherches !

QUESTIONS ?