

Chunking dans le cadre du projet ANR EPAC

- Principe général et fonctionnement -

Abdenour MOKRANE
Laboratoire d'Informatique (LI)
Université François Rabelais Tours

1. Contenu du répertoire EPAC

Le répertoire EPAC est composé de 4 sous répertoires :

- Le répertoire Corpus contient les corpus du projet EPAC et des fichiers de tests du fonctionnement.
- Le répertoire Documentation contient les manuels d'utilisation d'Unitex et CasSys, les articles publiés et les rapports internes, notamment pour le sous projet WP5 du projet EPAC.
- Le répertoire Tools contient les outils nécessaires (Unitex, CasSys, JRE) et les instructions d'installation. Il est donné dans ce même répertoire, à titre d'indication, un exemple de répertoire de travail Unitex (nommé RepTravailUnitex), il est inutile de recopier ce répertoire, il suffit seulement d'indiquer au démarrage d'Unitex le chemin d'un répertoire de travail (vide au départ) et Unitex se charge de la copie des fichiers nécessaires pour la langue sélectionnée.
- Le répertoire TransChunks contient la base des transducteurs et automates pour le chunking.

2. Principe général

Le chunking des corpus dans le cadre du projet EPAC est basé sur les cascades de transducteurs (lire les articles de recherche publiés à TALN 2008 et LREC 2008 pour plus de détails sur la méthodologie).

La base de transducteurs et automates du répertoire TransChunks est structuré comme suit.

- Les répertoires BTPos1, BTPos2 et BTPos3 contiennent les automates d'identification des mots suivant le format défini dans le rapport technique WP5_formats_echanges. Un chunk peut être composé d'un ou plusieurs mots, donc un mot peut être en début, fin ou entre le début et la fin d'un chunk. Un chunk est annoté par son token+word début, son token+word fin et un identificateur unique, donc il est nécessaire de conserver ces données.
 - Le répertoire BTPos1 contient les automates d'identification des mots (tags) en début des chunks (dans ce cas les word+token début des chunks sont conservés dans des variables Unitex).
 - Le répertoire BTPos2 contient les automates d'identification des mots (tags) qui sont entre le début et fin des chunks (dans ce cas il n'y a pas besoin de conserver des informations d'annotations des chunks).
 - Le répertoire BTPos3 contient les automates d'identification des mots (tags) en fin des chunks (dans ce cas les word+token fin des chunks sont conservés dans des variables Unitex).

NB. Un mot est composé de son identificateur unique, la référence à son token, son tag et le contenu (le mot lui-même).

En plus des word et token début des chunks, des informations sont conservées également pour l'identification unique des chunks.

NB. Dans le cas de chunks composés d'un seul mot (word+token début identiques au word+token fin), seuls les automates du répertoire BTPos1 sont utilisés.

Exemple 1.

La figure 1 illustre un exemple d'automate d'identification d'un mot étiqueté adverbe (ADV) et de récupération des informations nécessaires pour l'annotation des chunks, il s'agit d'un exemple d'automate du répertoire BTPos1, c'est le même principe pour les automates du répertoire BTPos3, pour BTPos2 la seule différence est l'absence de variables Unitex.

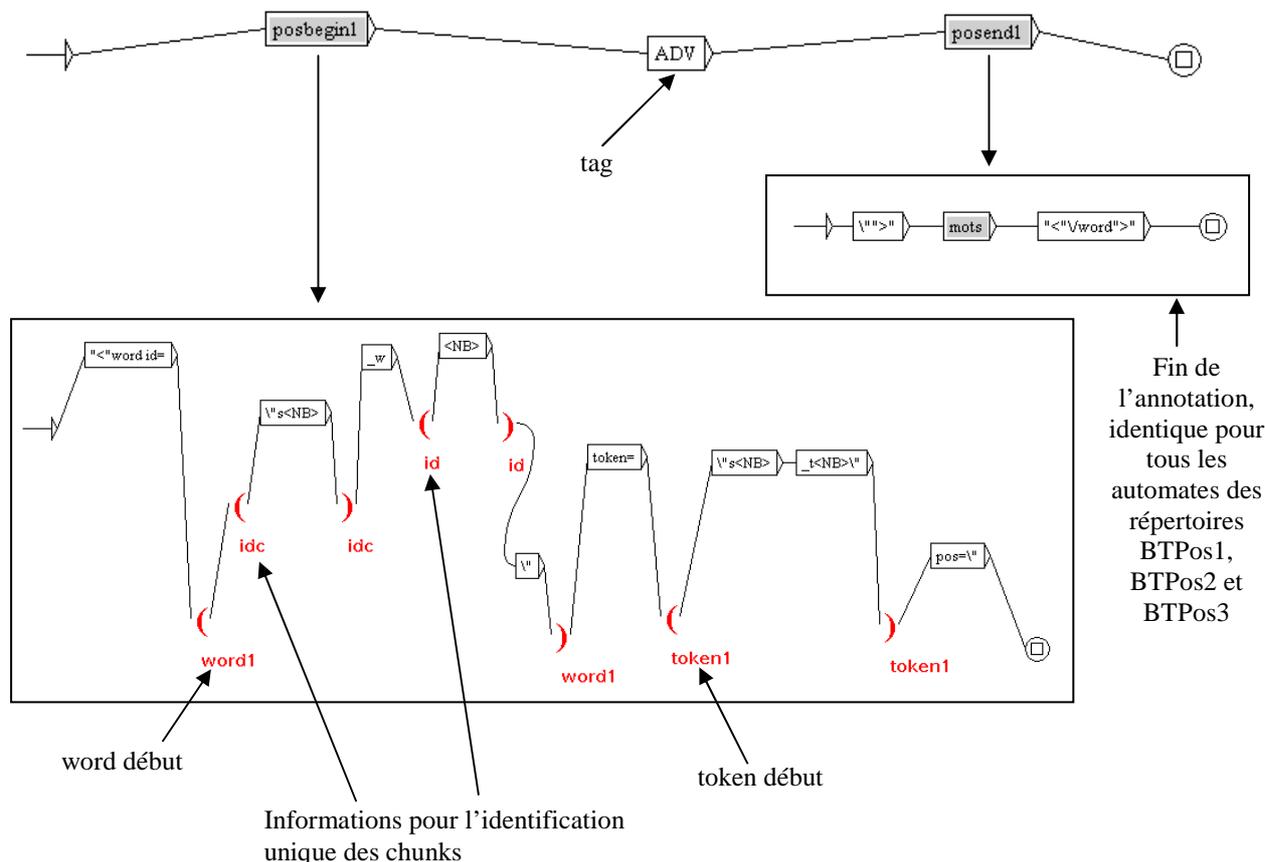


Figure 1. *exemple d'automate d'identification d'un mot*

➤ Les répertoires BTChunks1 et BTChunks2 contiennent les transducteurs d'annotation des sorties des chunks suivant également le rapport technique WP5_formats_echanges.

- Le répertoire BTChunks1 contient les transducteurs d'annotation des chunks composés d'un seul mot.
- Le répertoire BTChunks2 contient les transducteurs d'annotation des chunks composés de plusieurs mots.

NB. L'annotation des chunks se base sur les variables Unitex illustré dans l'exemple précédent.

Exemple 2.

La figure 2 illustre un exemple d'annotation d'un chunk nominal (GN) composé d'un seul mot. Il s'agit d'un exemple de transducteur du répertoire BTChunks1. C'est le même principe

pour l'annotation des chunks composés de plusieurs mots (répertoire BTChunks2) avec les word+token début et fin non identiques (variables Unitex différentes).

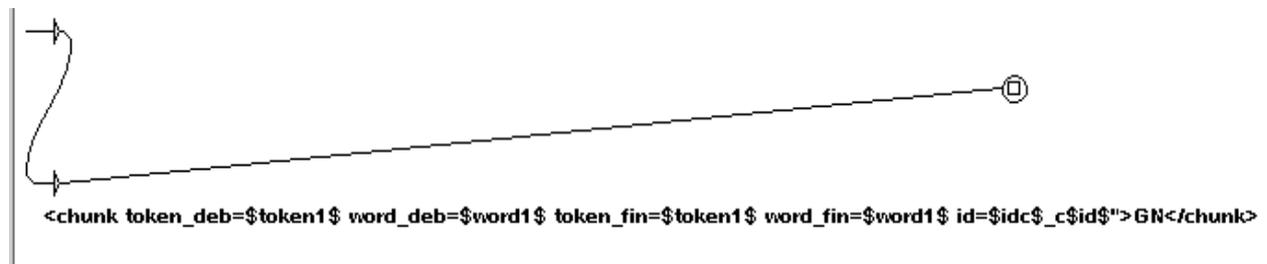


Figure 2. *exemple de transducteur d'annotation de chunk*

➤ Le répertoire TransChunks inclut également :

- Les transducteurs principaux d'identification des différents types de chunks, pour des raisons de lisibilité, certains transducteurs principaux sont découpés en sous graphes Unitex qui se trouvent également au même niveau que ces transducteurs. Pour plus d'informations sur les différents types de chunks et des exemples de transducteurs principaux, lire les articles publiés sur ce sujet disponibles dans le répertoire documentation. La figure 3 illustre un exemple de transducteur principal du chunk adverbial (GR).
- L'automate mots qui accepte toutes sortes de mots en fonction des formats des données dans le cadre du projet EPAC (extension de l'ensemble des motifs reconnus par le symbole spécial Unitex <MOT>, le sens des symboles Unitex est donné dans le manuel Unitex). Le transducteur motinc permettant de reconnaître des chunks GN composés de noms propres ou suite de noms propres non reconnus à la phase de Pos Tagging.
- Les fichiers de configurations CasSys : le fichier Chunks.conf de la cascade (tous les transducteurs principaux sont appelés dans ce fichier), les deux fichiers FormatAnalyse.conf et Format EPAC.conf qui concernent la génération des formats de sorties finales du chunking, l'utilisation de ces fichiers sera décrite dans la section suivante. Ces deux derniers fichiers font appel respectivement aux deux transducteurs formatanalyse (faisant appel aux deux sous graphes Unitex : anaypos et anychunk) et formatepac (faisant appel au sous graphe Unitex : anychunk).
- Une copie du programme exécutable CasSys (CasSys_xml.exe), ce même programme est disponible également dans le répertoire Tools.

CasSys soit sur un seul fichier (dans le cas d'un corpus composé d'un seul fichier) ou un répertoire de fichiers (dans le cas d'un corpus composé de plusieurs fichiers).
L'exemple 3 illustre d'une manière synthétique ces différentes opérations liées au chunking.

Exemple 3.

Soit le « CorpusTest.txt » un fichier à traiter disponible dans le répertoire Corpus, dans l'invite de commandes DOS en supposant qu'on se place dans le répertoire TransChunks :

1. Pour le lancement du chunking, lancer la commande suivante :

```
CasSys_xml chunks.conf ..\corpus\CorpusTest.txt
```

Les résultats de cette première opération sont stockés dans le sous répertoire « CorpusTest_csc » du répertoire Corpus.

2. Pour la génération du chunking au format EPAC, lancer le chunking sur le fichier CorpusTest_tag.snt disponible dans le répertoire « CorpusTest_csc » résultant de la première opération avec la commande suivante :

```
CasSys_xml FormatEPAC.conf ..\corpus\CorpusTest_csc\CorpusTest_tag.snt
```

Les résultats de cette deuxième opération sont stockés dans le sous répertoire « corpustest_tag_csc » du répertoire « CorpusTest_csc », le fichier corpustest_tag.idx contient le résultat du chunking au format EPAC.

Pour générer le chunking au format analyse, il suffit de lancer l'opération 2 avec le fichier de configuration CasSys « FormatAnalyse.conf ». Si on veut garder les deux formats, attention à ne pas écraser les données résultant en lançant les opérations liées aux formats.

Pour le lancement de l'opération 1 sur un répertoire de fichiers, il suffit de lancer CasSys avec l'option -R, soit « CorpusDoc » un répertoire de fichiers à traiter, si ce dernier est disponible dans le répertoire « Corpus », la commande suivante est à lancer :

```
CasSys_xml chunks.conf -R ..\Corpus\CorpusDoc
```

Le chunking est donc lancer sur tous les fichiers du répertoire « CorpusDoc », pour la génération du chunking dans les formats voulu, il suffit de lancer l'opération 2. De la même manière, il est possible également de regrouper dans un seul répertoire tous les fichiers tag résultant de l'opération 1 et lancer l'opération 2 sur ce répertoire.

4. Format de fichiers en entrée et sortie

Il est à noter que les fichiers à traiter doivent être au format Unicode. Consulter le manuel CasSys pour plus d'informations sur le fonctionnement et les données en entrée et en sortie.

Si vous avez à travailler sur des fichiers ASCII, Unitex dispose d'un outil assurant la conversion de ces derniers au format Unicode. Deux manières de procéder pour cela :

- via l'interface d'Unitex : Menu `File Edition -> Transcode Files`
- en ligne de commande : directive `convert`.

Syntaxe de la directive de conversion en ligne de commande :

```
Convert [OPTIONS] <text_1> [<text_2> <text_3> ...]
```

OPTIONS

- . -s X/--src=X codage en entrée
- . -d X/--dest=X codage en sortie (par défaut=LITTLE-ENDIAN);

Output options

- . -r/--replace les sorties écrasent les fichiers d'entrée (par défaut)
- . --ps=PFX les fichiers d'entrée sont renommés avec le préfixe PFX
- . --pd=PFX les fichiers de sortie sont renommés avec le préfixe PFX

5. Bugs connus

Liste des bugs identifiés à ce jour avec la version Unitex 1.2 (SECARE ne fonctionne de manière sécurisée qu'avec cette version).

- SECARE/CasSys doit être lancé depuis le répertoire Transchunks où sont situés les fichiers transducteurs. Si cette contrainte n'est pas respectée, message d'erreur « Cannot find .\TransChunks ».
- SECARE/CasSys rend la main à la fin de la cascade si aucun motif n'a été détecté par les transducteurs. Situation plus que hautement improbable (il faudrait qu'absolument aucun chunk ne soit détecté dans votre document).
- SECARE/CasSys rend la main au début de l'application du premier transducteur si le nom de votre fichier d'entrée présente une extension comportant plusieurs préfixes séparés par plusieurs points. Par exemple : toto.tag.xml (erreur due à un problème de nommage lors de la création du répertoire de sortie). Il est donc recommandé de renommer vos fichiers, par exemple ici : toto_tag.xml.