TALN 2010, Montréal, 19–23 juillet 2010

MACAON
Une chaîne linguistique pour le traitement de graphes de mots

Alexis Nasr Frédéric Béchet Jean-François Rey *
Laboratoire d'Informatique Fondamentale de Marseille
Université Aix-Marseille
(alexis.nasr,frederic.bechet,jean-francois.rey)@lif.univ-mrs.fr

1 Introduction

MACAON est une chaîne de traitement linguistique séquentielle composée de plusieurs modules réalisant des traitements classiques (découpage en mots, étiquetage morpho-syntaxique, lemmatisation, analyse morphologique, analyse syntaxique partielle) sur une entrée textuelle native (produite par un être humain) ou sur des hypothèses multiples de mots, produits automatiquement. MACAON est distribué sous licence GPL et téléchargeable à partir de l'adresse `macaon.lif.univ-mrs.fr`.

D'un point de vue général, un module MACAON peut être vu comme un module d'annotation[1] qui ajoute à son entrée un niveau d'annotation qui dépend généralement des annotations produites par les modules précédents. Les différents modules communiquent entre eux par l'intermédiaire de fichiers XML qui permettent de représenter les différents niveaux d'annotation. De plus, la chaîne permet de conserver la structuration initiale XML des documents traités (structuration logique d'un document, informations provenant d'un système de dialogue ...).

Une des principales caractéristiques de MACAON est la prise en compte d'ambiguïtés en entrée et en sortie des différents modules. La majorité des modules de MACAON acceptent des entrées ambiguës (plusieurs hypothèses d'annotation) et produisent à leur tour des sorties ambiguës, de manière à repousser le plus tard possible la tâche de désambiguïsation. La représentation compacte des structures ambiguës constitue un des fondements du format d'échange MACAON, décrit en 2. De plus, chaque module peut pondérer les solutions qu'il construit et limiter l'ambiguïté produite à un nombre donné de solutions, déterminé par l'utilisateur.

2 Le format d'échange de données

Le format d'échange de données repose sur quatre concepts, celui de *segment*, d'*attribut*, de *niveau* et de *segmentation*.

*Ces travaux sont en partie financés par les projets ANR Sequoia ANR-08-EMER-013 et DECODA 2009-CORD-005-01

[1] Nous regroupons sous le vocable annotation aussi bien des tâches d'étiquetage que de segmentation ou de construction d'objets plus complexes, tel que des arbres syntaxiques.

Un *segment* est, comme son nom l'indique, un segment du texte ou du signal de parole à traiter, tel qu'une phrase, une proposition, un constituant syntaxique, une unité lexicale, une entité nommée ...Un segment possède des attributs qui permettent d'en décrire différents aspects. Un constituant syntaxique, par exemple, définira l'attribut *type* qui précisera le type du syntagme (constituant verbal, nominal ...). Un segment est composé d'un ou plusieurs segments plus simples.

Une séquence de segments qui couvre l'integralité d'une phrase dans le cas de l'écrit ou d'un tour de parole dans le cas de l'oral, est appelé *segmentation*. Cette dernière peut être affecté d'un poids.

Un *niveau* d'analyse regroupe des segments d'une nature donnée, ainsi que des segmentations définies sur ces segments. Quatre niveaux sont distingués pour le moment dans MACAON : le niveau pré-lexical, le niveau lexical, le niveau morpho-syntaxique et le niveau syntaxique.

Les différents segments sont liés entre eux par deux types de relations, la relation de précédence qui organise les segments linéairement pour constituer des segmentations et la relation de dominance qui décrit comment un segment se décompose en segments plus petits du même niveau ou d'un niveau inférieur.

Nous avons représenté ci-dessous, sous la forme d'un tableau, un exemple d'analyse en quatre niveaux de quelques hypothèses d'un graphe de parole qui pourraient être produites sur l'entrée *Jean mange une pomme de terre*. Le tableau permet de visualiser les quatre niveaux d'analyse, les segments (un segment par case) ainsi que les relations de précédence (le segment pré-lexical *Jean* peut être suivi du segment *lange* ou du segment *mange*) et de dominance (le segment lexical *pomme de terre* domine les trois segments pré-lexicaux *pomme*, *de* et *terre*).

<table>
<tr><td rowspan="3">syntaxique</td><td></td><td>SN</td><td colspan="2">SN</td><td colspan="3">SV</td></tr>
<tr><td colspan="4"></td><td colspan="3">SP</td></tr>
<tr><td>SN</td><td>SN</td><td colspan="5">SN</td></tr>
<tr><td rowspan="3">morpho-syntaxique</td><td rowspan="3">np</td><td>v</td><td rowspan="3">det</td><td colspan="4">nc</td></tr>
<tr><td>nc</td><td>nc</td><td colspan="3">v</td></tr>
<tr><td colspan="2"></td><td colspan="2">prep</td><td>nc</td></tr>
<tr><td rowspan="3">lexical</td><td rowspan="3">Jean</td><td>mange</td><td rowspan="3">une</td><td colspan="4">pomme de terre</td></tr>
<tr><td>lange</td><td>tome</td><td colspan="3">déterre</td></tr>
<tr><td></td><td>pomme</td><td>de</td><td>terre</td></tr>
<tr><td rowspan="2">pré-lexical</td><td rowspan="2">Jean</td><td>lange</td><td rowspan="2">une</td><td>tome</td><td colspan="3">déterre</td></tr>
<tr><td>mange</td><td>pomme</td><td>de</td><td>terre</td></tr>
</table>

Cet exemple nous permet d'illuster les différents cas d'ambiguïté pris en compte ainsi que leur mode de représentation.

Le cas d'ambiguïté le plus immédiat est celui de l'ambiguïté de segmentation : différentes segmentations sont possibles à chaque niveau. Cette ambiguïté est représentée de manière compacte en factorisant les segments communs à plusieurs segmentations sous la forme d'un automate fini.

Le second cas d'ambiguïté est celui de l'ambiguïté de dominance, où un segment peut se décomposer de différentes façons en segments de niveau inférieur. Un tel cas apparaît dans l'exemple précédent, où le nom commun (nc) du niveau morpho-syntaxique (représenté en gras), domine les deux segments lexicaux *pomme* et *tomme*.

Mise en œuvre en XML

Comme nous l'avons annoncé dans l'introduction, le format d'échange MACAON est implémenté en XML. Un segment est représenté par une balise `<segment>`. Cette dernière possède quatre attributs obligatoires :

- `type` indique le type du segment, quatre types de segments sont définis pour l'instant :`atome` (unité pré-lexicale), `ulex` (unité lexicale), `cat` (partie de discours) et `chunk` (groupe syntaxique non récursif).
- `id` permet d'attribuer au segment un identifiant unique au sein du document, afin de pouvoir y faire référence.
- `start` et `end` permettent de préciser le début et la fin du segment. Il s'agit de valeurs numériques qui peuvent faire référence à la position du premier et du dernier caractères du segment dans la chaîne textuelle ou au temps de début et de fin du segment dans le signal de parole.

Un segment peut posséder tout autre attribut jugé utile pour un certain niveau de description. On trouve en particulier souvent l'attribut `stype` qui permet d'affiner la valeur de l'attribut `type`. La relation de dominance est représentée par l'enchâssement de balises XML. Illustrons cela sur l'exemple donné précédemment, où l'unité lexicale *pomme de terre* regroupe les trois atomes *pomme*, *de* et *terre*, ayant respectivement pour `id` les valeurs `a1`, `a2` et `a3`, au sein d'une séquence de 3 éléments (dans une balise `<sequence>`).

```
<segment type=”ulex” id=”u1”> <sequence> <elt segref=”a1”/> <elt segref=”a2”/>
<elt segref=”a3”/> </sequence> </segment>
```

Le cas précédemment évoqué où la partie de discours *nc* domine les deux unites lexicales *pomme* et *tome*, ayant respectivement pour `id` `u3` et `u4`, est représenté par une disjonction de séquences à l'intérieur d'un segment :

```
<segment type=”cat” stype=”nc” id=”c1”> <sequence> <elt segref=”u3” w=”3.37”/>
</sequence> <sequence> <elt segref=”u4” w=”4.53”/> </sequence> </segment>
```

La relation de dominance est associée à un poids matérialisé par la balise `w`. Ce poids permet de représenter dans l'exemple précédent la probabilité d'une unité lexicale étant donné une catégorie, telles qu'on les trouve dans un étiqueteur morpho-syntaxique à base de chaînes de Markov cachées.

Comme nous l'avons déjà mentionné, les segmentations sont représentées sous la forme d'automates finis pondérés. Ces derniers sont représentés classiquement comme une série de transitions entre états plus la spécification d'un état intial et des états d'acceptation, comme dans l'exemple ci-dessous :

```
<fsm n="9"> <start n="0"/> <accept n="6"/> <ltrans>
<trans o="0" d="1" i="a1" w="7.23"/> <trans o="1" d="2" i="a2" w="9.00"/>
<trans o="1" d="2" i="a3" w="3.78"/> <trans o="2" d="3" i="a4" w="7.37"/>
<trans o="3" d="4" i="a5" w="3.73"/> <trans o="3" d="4" i="a6" w="6.67"/>
<trans o="4" d="5" i="a7" w="4.56"/> <trans o="5" d="6" i="a8" w="2.63"/>
<trans o="4" d="6" i="a9" w="7.63"/> </ltrans> </fsm>
```

La balise `<trans/>` matérialise une transition, les champs `o,d,i` et `w` représentent respectivement l'état d'origine de la transition, son état destination, son étiquette (une référence à un segment) et un poids.

Finalement, un niveau d'analyse est matérialisé par la balise `<section>` qui comporte une balise `<segments>` qui regroupe tous les segments correpondant à ce niveau d'analyse et une balise `<fsm>` qui représente l'ensemble des segmentations.

3 Les modules

Différents modules standard ont été développés dans le cadre de MACAON, ils sont brièvement decrits ci-dessous. Tous les modules partagent un certain nombre de caractéristiques : ils vérifient tous, bien entendu, le format d'échange décrit ci-dessus. Ils laissent tous, sous la forme d'une balise `<maca_stamp>` une trace de leur passage dans le fichier traité et reconnaissent un jeu d'options standard.

maca_select est un module de pré-traitement, il parcours un fichier XML et insère des balises `<macaon>` sous les balises spécifiées par l'utilisateur. Les modules suivants ne traiteront que les parties textuelles comprises dans des balises `<macaon>`.

maca_segmenter réalise la segmentation du texte en phrases en fonction du contexte dans lequel se trouve un signe de ponctuation.

maca_tokenizer réalise le découpage d'une phrase en unités pré-lexicales. Il repose sur une grammaire régulière qui définit un ensemble de types d'atomes. Un analyseur lexical détecte les séquences de caractères (en fonction de la grammaire) et leur associe un type.

maca_lexer permet le regroupement d'unités pré-lexicales en unités lexicales. Il repose sur le Lexique des Formes Fléchies du Français (le*fff* : http ://atoll.inria.fr/~sagot/lefff.html). Il implémente un algorithme de programmation dynamique qui construit toutes les segmentations possibles en unités lexicales.

maca_tagger associe à toute unité lexicale une ou plusieurs parties de discours. Il repose sur une chaîne de Markov cachée implémentant un modèle trigramme entrainé sur le corpus French Treebank (http ://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php).

maca_anamorph réalise l'analyse morphologique d'unités lexicales associées à une catégorie morpho-syntaxique. Les informations morphologiques proviennent du le*fff*.

maca_chunker regroupe des séquences de parties de discours en unités syntaxiques non récursives.

maca_conv permet de convertir au format MACAON des graphes de mots produits par des systèmes de transcription automatiques représentés au format HTK (htk.eng.cam.ac.uk) ou
FSM (www2.research.att.com/~fsmtools/fsm).

maca2txt permet une visualisation d'un fichier MACAON, au format textuel.

maca_view est une interface graphique permettant de visualiser des fichiers MACAON et de lancer la chaîne de traitement sur un fichier.

4 Conclusion

Nous avons présenté dans cet article la chaîne de traitement MACAON qui permet de traiter aussi bien du texte natif que des hypothèses lexicales multiples produites automatiquement. Plusieurs évolutions de MACAON sont en cours, tel que l'ajout de nouveaux modules (analyse syntaxique partielle en dépendance, détecteur d'entité nommées) et l'évolution du format d'échange afin de représenter des segments non contigus.