LOL : Langage objet dédié à la programmation linguistique

Jimmy Ma, Mickaël Mounier, Helena Blancafort, Javier Couto, Claude de Loupy

(1) Syllabs, 15 rue Jean-Baptiste Berlier, 75013 Paris, France
{ma, mounier, blancafort, couto, loupy}@syllabs.com

LOL (*Linguistic Object Language*) est un langage dédié à la description d'objets linguistiques développé par la société Syllabs. Ce langage s'intègre dans une plateforme industrielle et permet aux linguistes d'écrire des règles d'extraction d'information ainsi que des règles de correction d'étiquetage morphosyntaxique. Lors de la conception de ce langage, l'idée était de proposer un vrai langage de programmation qui soit à la fois puissant au niveau de l'expression et à la fois simple à utiliser par des linguistes. De plus, ce langage permet une manipulation de plus haut niveau sans nuire aux performances du système produit.

LOL est un langage déclaratif qui permet de visualiser la langue sous la forme d'un langage de description de connaissances linguistiques plutôt que d'un langage de programmation. LOL est aussi un langage objet avec des objets prédéfinis comme les *tokens* (mots, préfixes, etc.), les phrases, etc. Les linguistes peuvent définir leurs propres objets, des listes d'éléments, ou des objets plus complexes reconnus à l'aide de patrons. Les spécifications écrites par les linguistes sont interprétées et mises en relation avec l'ensemble des ressources et outils de base développés par Syllabs (un lexique morphosyntaxique, un segmenteur, un étiqueteur morphosyntaxique, un *guesser*). Cette plateforme inclut également un outil de visualisation en html. La sortie peut actuellement être fournie sous format txt, XML ou JSON. Les analyseurs ainsi produits sont mis à disposition via des APIs REST (web services) sur les plateformes de Syllabs.

Concernant la manipulation des objets linguistiques, le linguiste manipule des *tokens* et ses différents attributs. Il peut ainsi accéder à différentes informations du *token*: 1) classe du *token* (mot, url, etc.), 2) information lexicales et morphosyntaxiques, 3) information graphique (typographie, nombre de caractères, etc.), 4) informations sur le *guessing* (s'il s'agit d'un *token* inconnu et si oui, s'il a été deviné) ; 5) positionnelles pour faire des conditions en fonction de la position dans le texte (ex : début de phrase ou de paragraphe). Ci-dessous un exemple de règle de correction et de règle d'extraction.

```
correction_rule  {
// correction d'erreur
//due à l'ambiguïté Nom-Adjectif
        [conditions]
        token.POS(D) | token.POS(X)
        token.POS(A)& token.ambig(N)
        token.POS(Sp)
        [actions]
        match[2].POS=N
}
```

```
extraction_rule {
[conditions]
!left_filter_potentialPN | token.class(begin)
f: FirstNameCap{1,2}
l: (PREMOD_NOM)? FamilyNameCap //(token.string("-")? FamilyNameCap)?
[actions]
        create Person[f,l] : priority(1); confidence = 1.0 {
                firstname = match[f]
                lastname  = match[l]
        }
}
```

Figure 1 : Exemples de règles

Aujourd'hui LOL est utilisé dans plusieurs applications industrielles commercialisées, notamment pour des applications basée sur l'extraction d'information (par ex. analyse de tonalité) et pouvant s'intégrer dans un processus de veille ou de tagging automatique de textes. Le linguiste peut également utiliser l'outil pour améliorer l'étiqueteur morphosyntaxique, voire le spécialiser sur le corpus du client en jouant sur les différentes propriétés du token (par ex. : conditions d'application d'une règle en fonction de la classe du token, la position dans le texte, le contexte et graphie). LOL est indépendant de la langue et est utilisé dans des applications industrielles ou de recherche en 8 langues, dont le chinois et le russe. Nous prévoyons d'ouvrir la plateforme de manière à permettre à des utilisateurs de créer leurs propres analyseurs.