ACOLAD, Plateforme pour l'édition collaborative dépendancielle

Francis Brunet-Manquat, Jérôme Goulian

▶ **To cite this version:**

Francis Brunet-Manquat, Jérôme Goulian. ACOLAD, Plateforme pour l'édition collaborative dépendancielle. Actes de la conférence conjointe JEP-TALN-RECITAL 2012, Jun 2012, Grenoble, France. pp.335-342. hal-00953820

HAL Id: hal-00953820
https://hal.science/hal-00953820

Submitted on 31 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ACOLAD
Plateforme pour l'édition collaborative dépendancielle

Francis Brunet-Manquat et Jérôme Goulian
LIG-GETALP, Université Pierre Mendès France Grenoble 2
Francis.Brunet-Manquat@imag.fr et Jerome.Goulian@imag.fr

RESUME

Cet article présente une plateforme open-source pour l'édition collaborative de corpus de dépendances. Cette plateforme, nommée ACOLAD (Annotation de COrpus Linguistique pour l'Analyse de Dépendances), propose des services manuels de segmentation et d'annotation multi-niveaux (segmentation en mots et en syntagmes minimaux (chunks), annotation morphosyntaxique des mots, annotation syntaxique des chunks et annotation syntaxique des dépendances entre mots ou entre chunks). Dans cet article, nous présentons la plateforme ACOLAD, puis nous détaillons la représentation pivot utilisée pour gérer les annotations concurrentes, enfin décrivons le mécanisme d'importation de ressources linguistiques externes.

ABSTRACT

ACOLAD: platform for collaborative dependency annotation

This paper presents an open-source platform for collaborative editing dependency corpora. ACOLAD platform (Annotation of corpus linguistics for the analysis of dependencies) offers manual annotation services such as segmentation and multi-level annotation (segmentation into words and phrases minimum (chunks), morphosyntactic annotation of words, syntactic annotation chunks and annotating syntactic dependencies between words or chunks). In this paper, we present ACOLAD platform, then we detail the representation used to manage concurrent annotations, then we describe the mechanism for importing external linguistic resources.

MOTS-CLES : annotation collaborative de corpus, annotations concurrentes, dépendances
KEYWORDS : corpus collaborative annotation, concurrent annotations, dependencies

1 Introduction

De nombreuses applications de TAL nécessitent de grandes quantités de données annotées manuellement. La production de ces données est coûteuse. D'autre part, la nature et la qualité des annotations à produire dépendent très largement des besoins en terme d'exploitations futures du corpus (Valli et Véronis, 1999). Pour faciliter la production de tels corpus, plusieurs outils récents ont été développés parmi lesquels on peut citer : l'application Web 2.0 *System EasyRef* développé dans le cadre de l'ANR action Passage pour annoter des corpus syntaxiques dans les formats Easy et Passage (Paroubek et al., 2009), l'extension firefox *WebAnnotator* qui permet d'annoter des pages Web selon une DTD définie par l'utilisateur (Xavier, 2012). Ces outils tentent de résoudre de nombreux problèmes liés à la création de corpus, en particulier, l'aspect collaboratif pour *EasyRef* et l'aspect générique pour *WebAnnotator*. Dans cet article, nous nous intéressons à 2 problèmes cruciaux liés à l'annotation de corpus : Comment

représenter les annotations concurrentes ? Comment importer et utiliser de manière générique des ressources linguistiques *externes* comme des dictionnaires ou des analyses morphosyntaxiques ?

Nous tentons avec la plateforme ACOLAD (Annotation de COrpus Linguistique pour l'Analyse de Dépendances) de répondre à ces questions. Cette plateforme open-source a été développée avec pour objectif de faciliter la tâche d'édition collaborative lors de la création d'un corpus de dépendance. Elle propose des services manuels de segmentation et d'annotation multi-niveaux (segmentation en mots et en syntagmes minimaux (chunks), annotation morphosyntaxique des mots, annotation syntaxique des chunks et annotation syntaxique des dépendances entre mots ou entre chunks).

Après une vue d'ensemble de la plateforme ACOLAD, nous nous focalisons dans cet article sur la représentation pivot utilisée pour gérer les annotations concurrentes et le mécanisme d'importation de ressources linguistiques externes.

2 ACOLAD, une plateforme pour l'édition de corpus de dépendances

2.1 Présentation

L'idée est de proposer sur la même plateforme une palette de services permettant la création de corpus annotés pour les besoins classiques de développement d'outils linguistiques. Les intérêts de notre environnement sont les suivants :

– visualisation et édition graphique simple des annotations (voir figure 1) ;

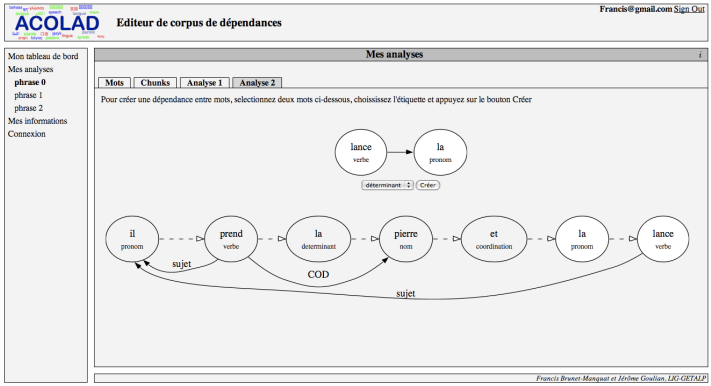

Figure 1 : création d'une analyse de dépendances

– édition manuelle configurable :
  - choix entre différentes granularités d'unités d'annotations (dépendances entre mots et/ou entre chunks) ;
  - possibilité d'utiliser différents jeux d'étiquettes (grammaticales, syntaxiques au niveau des chunks et des dépendances) pour tenir compte des spécificités de chaque corpus (écrit, oral transcrit ou oral issu de la reconnaissance de parole par exemple) et des besoins en terme d'exploitation future du corpus ;

- choix des contraintes structurelles de dépendances à associer à l'édition en cours[1] (projectivité ou non projectivité, analyse totale ou partielle) ;

– possibilité d'éditer simultanément les ambiguïtés d'analyse tant au niveau mots, chunks, qu'au niveau des dépendances syntaxiques.

2.2 Architecture de la plateforme ACOLAD

La plateforme ACOLAD est une application web consacrée au développement collaboratif de corpus de dépendances. La plateforme est organisée en une architecture 3-tiers classique (voir figure 2) : une couche de *présentation* (responsable de l'interface avec les utilisateurs), une couche de traitement (qui fournit les services) et une couche *données* (responsable du stockage des données persistantes).

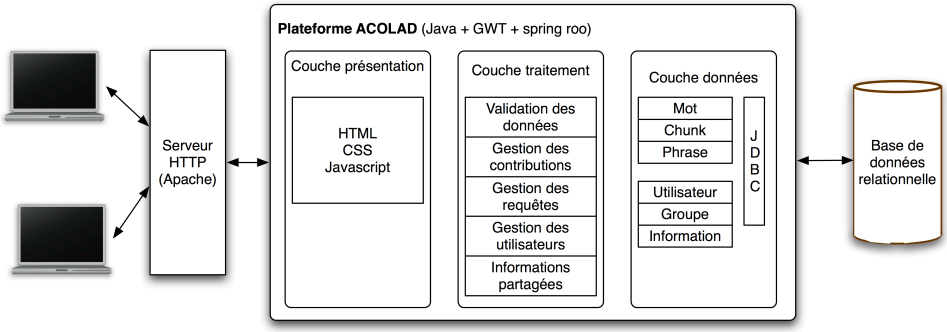

Figure 2 : architecture générale de la plateforme ACOLAD

Pour utiliser la plateforme ACOLAD, le responsable de la tâche d'annotation du corpus n'a pas à écrire de code spécifique Java ni de pages Web dynamiques spécifiques. Il doit simplement fournir, outre la liste des annotateurs et des modérateurs, une description du jeu d'étiquettes (sous forme XML) et éventuellement les dictionnaires nécessaires à la tâche (quelques dictionnaires sont déjà intégrés, par exemple LEFFF/DELAF pour le français).

3 Représentation pivot pour l'annotation concurrente

La plateforme ACOLAD se base sur une représentation par pivot pour distinguer les différences de segmentation, d'étiquetage et d'annotation dépendancielle.

3.1 Matrice de dépendances (MD)

Toute annotation est décrite, dans notre plate-forme, par une représentation matricielle. Notre représentation, nommée matrice de dépendance (MD), est un couple <L, M> composé de :

– Une liste de nœuds (L), un nœud étant composé d'informations linguistiques

[1] Pour le moment il s'agit d'une liste pré-établie de contraintes.

relatives aux mots et/ou aux chunks[2] qu'il décrit ;

– Une matrice carrée (M) permettant de décrire les dépendances entre nœuds (implémentée sous forme de matrice creuse). La case (i, j) contient l'ensemble des dépendances entre le nœud i et le nœud j de la liste de nœuds.

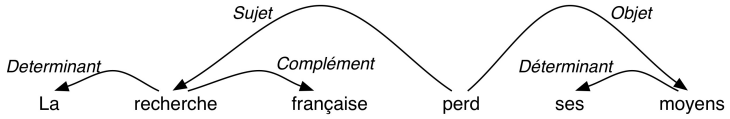

Figure 3 : exemple d'analyse de dépendances

La MD correspondant à la structure de dépendance syntaxique de la figure 3 est :

L =

la	::	cat=déterminant
recherche	::	cat=nom
française	::	cat= adjectif
perd	::	cat=verbe
ses	::	cat=déterminant
moyens	::	cat=nom

M =

relation	la	recherche	française	perd	ses	moyens
la						
recherche	Déterminant		Complément			
française						
perd		Sujet				Objet
ses						
moyens					Déterminant	

Cette représentation matricielle des données présente trois avantages pour le traitement informatique :

– Maniabilité : de nombreux outils mathématiques sont associés aux matrices : ajout, suppression, comparaison, etc. Tous ces outils permettent un traitement simple de l'information contenue dans une matrice.
– Efficacité : les méthodes utilisant les matrices comme structures de données, telle que la reconnaissance de motifs ou la fusion de matrice, se montrent très efficaces et très simples à mettre en place.
– Simplicité de la description d'analyse multi-niveaux (toutes les informations peuvent être présente dans la même matrice de dépendances)

3.2 Annotations concurrentes

Pour illustrer le mécanisme de gestion d'annotations concurrentes d'ACOLAD, nous prenons comme exemple dans la suite le cas de segmentations multiples. Le principe consiste à regrouper les nœuds représentant la même segmentation dans la phrase (information commune minimale). Mais elle consiste également à représenter les discordances issues des différentes segmentations des structures et dues, par exemple, aux mots composés, aux entrées des dictionnaires (États Unis ou États-Unis), etc.

Pour ce faire, nous créons une structure, appelée réseau de segmentation (RS), représentant les différentes segmentations de la phrase et permettant de lier les nœuds

[2] Dans la suite de l'article, on ne traitera pour l'exemple que les relations de dépendance entre mots.

des structures. Ce réseau peut être vu comme un « pivot de liaison » entre ces structures. Ce réseau est un treillis, chaque nœud du réseau représentant une segmentation possible d'un mot et servant de liaison entre les nœuds des structures de dépendance. Concrètement, un nœud Nrs d'un RS contient deux informations :

- SNODE(Nrs) : intervalles représentant la sous-chaîne dans la phrase correspondant au nœud Nrs, Par exemple, les mots de la phrase « On avait dénombré cent vingt-neuf candidats » auront pour intervalles : On[1-2], avait[3-7], dénombré[8-15], etc. Cette information est basée sur les SSTC (Structured String-Tree Correspondences) proposée par (Boitet et Zaharin, 1988) ;
- L : un ensemble contenant les nœuds des structures liés au nœud Nrs.

Le réseau de segmentation final obtenu représente les segmentations possibles et lie les nœuds des structures entre eux. Une fois que la correspondance entre les nœuds des structures est établie, la plateforme peut fusionner ces structures pour fournir une unique représentation de dépendance combinant toutes les informations linguistiques relatives aux structures (illustré dans la figure 4).

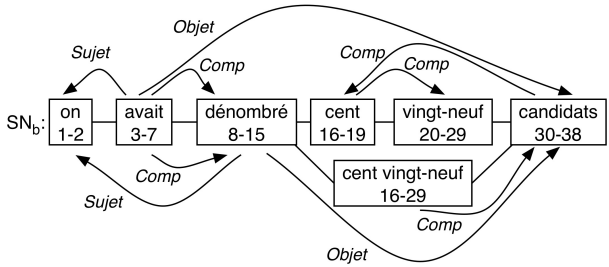

Figure 4 : exemple d'annotations concurrentes

4 Importer des informations linguistiques *externes*

La plate-forme ACOLAD doit être capable de récupérer des informations linguistiques provenant d'autres outils (par exemple d'analyseurs lexicaux ou morphosyntaxiques), c'est-à-dire d'extraire les informations des résultats qu'ils produisent puis de les interpréter. Nous pensons en effet que cette fonctionnalité doit faire partie intégrante de l'outil afin de fournir un cadre simple de transformation des données aux utilisateurs diffusable en tant que « plugin » avec l'outil. Le processus se compose de deux phases :

- La phase d'extraction proprement dite dépendante des analyseurs utilisés ;
- La phase de projection des informations extraites.

4.1 Extraction des informations

Un module d'extraction peut être généré facilement pour chaque outil externe. Ce module permet de lire les fichiers résultats produits, de vérifier leurs bonnes structurations et d'extraire les données linguistiques contenues dans ces fichiers.

4.1.1 Production des modules d'extraction

La production de modules d'extraction s'organise en deux étapes. Dans un premier temps, il faut définir un fichier de spécifications qui décrit le format du fichier résultat produit par l'outil (une grammaire de description). Il faut ensuite enrichir cette grammaire avec des méthodes[3] qui permettent d'extraire les données linguistiques contenues dans les résultats d'analyse. Un module d'extraction peut alors être produit à partir de la grammaire et des méthodes.

Chaque module d'extraction généré est constitué de deux analyseurs :

- Un analyseur lexical qui convertit une séquence de caractères provenant du fichier résultat à analyser en une séquence d'objets, nommés tokens ;
- Un analyseur syntaxique qui consomme la séquence de tokens calculée précédemment pour vérifier la syntaxe du fichier résultat. Il exécute également les actions associées à la grammaire.

4.1.2 Grammaire de description

Une grammaire de description représente le format des fichiers résultats fournis par un analyseur. Cette grammaire est de type hors contexte et elle est composée de deux ensembles :

- Un ensemble de tokens, représentant les objets (nombre, chaîne, etc.) contenus dans un fichier résultat ;
- Un ensemble de règles syntaxiques, représentant la syntaxe d'un fichier résultat en fonction des tokens.

4.1.3 Grammaire de description enrichie

Pour extraire les données linguistiques présentes dans les fichiers résultats, des méthodes d'extraction sont introduites dans les grammaires de description. Ces méthodes se présentent sous la forme de procédures insérées dans les règles syntaxiques. Elles permettent d'extraire les données linguistiques. Quand une règle syntaxique de la grammaire est appliquée, les méthodes correspondantes sont exécutées. La grammaire ainsi créée est nommée grammaire de description enrichie. Nous donnons ci-après un extrait d'une grammaire de description augmentée pour une analyse de dépendance.

```
// Règle syntaxique
void RelationDeDependance() {
   // Déclaration des variables
   String etiquette ; Mot mot1, mot2 ;

   etiquette=<Chaine> "(" mot1=Element() "," mot2=Element() ")"
      // où   - Element() est une règle syntaxique retournant un objet Mot
      //      - <Chaine> est un token représentant une chaîne
      {
         // Extraction d'une relation syntaxique avec la
         // méthode d'extraction AjoutRelationSyntaxique
         phraseCourante.AjoutRelationSyntaxique(etiquette, mot1, mot2)
```

[3] Un panel de méthodes est proposé pour réaliser l'extraction des informations linguistiques.

```
    }
}
void Element() {
   <Chaine> "^" <Chaine> "^" ( <Chaine> )+ ( "_" <Chaine> ) )? ":" <Nombre>
      // <Nombre> est un token représentant un nombre positif
      // le + signifie que la chaîne peut apparaître plusieurs fois
      // le ? signifie que la chaîne peut apparaître au moins une fois
}
```

La règle syntaxique précédente s'applique sur chacune des relations de dépendance extraites de l'analyse faite par l'analyseur XIP de la phrase « La recherche française perd ses moyens. » :

```
SUBJ(<perd^perdre^+VERB_P3SG:3>,<recherche^recherche^+NOUN_SG:1>)
VARG(<perd^perdre^+VERB_P3SG:3>,<moyens^moyen^+NOUN_PL:5>)
NN(<recherche^recherche^+NOUN_SG:1>,<française^française^+NOUN_INV:2>)
DETERM(<La^le^+DET_SG:0>,<recherche^recherche^+NOUN_SG:1>)
```

4.2 Projection des données extraites

Après avoir extrait toutes les données linguistiques des résultats d'un analyseur, il faut faire correspondre ces données à une norme commune établie, spécifiée en amont en fonction des jeux d'étiquettes donnés pour la tâche d'annotation.

Cette phase peut être reproduite facilement autant de fois que nécessaire et ainsi s'adapter aux différentes granularités des jeux d'étiquettes envisagés en fonction des besoins de la tâche.

Pour normaliser les données extraites, un ensemble de règles de projection est défini. Une règle de projection est constituée d'une partie gauche représentant les données linguistiques à reconnaître et d'une partie droite représentant les mêmes informations normalisées. Par exemple, pour les données brutes extraites dans l'exemple précédent, les règles instanciables sont[4] :

```
SUBJ($var1, $var2) ::= Sujet($var1, $var2)
VARG($var1, $var2) ::= Objet($var1, $var2)
NN($var1, $var2) ::= Complement($var1, $var2)
DETERM($var1, $var2) ::= Determinant($var2, $var1)
              // La relation DETERM est projetée en une relation
              // Determinant en modifiant l'ordre relatif des mots
```

L'extraction d'information d'ACOLAD est basée sur les outils développés pour la plateforme de combinaison d'analyseurs syntaxiques DepAn (Brunet-Manquat, 2005). L'approche a été expérimentée et validée dans (Brunet-Manquat, 2004).

5 Perspectives

ACOLAD propose des services manuels permettant de segmenter une phrase ou de produire des analyses de dépendances. Cet environnement est actuellement utilisé pour produire des analyses de dépendances pour le français. Mais cet outil, par son

[4] une variable $vari représente soit un mot soit un chunk

formalisme de dépendances et son aspect configurable (choix des jeux d'étiquettes, choix des contraintes structurelles de dépendances -- en particulier pour tenir compte de la variabilité de l'ordre des mots selon la langue (Holan, 2000)), a un potentiel multilingue.

Des expérimentations sont actuellement menées pour intégrer les premiers résultats d'ACOLAD dans nos travaux sur l'analyse syntaxique automatique et la production de documents auto-explicatifs (Blanchon et al., 2006).

Les services proposés par ACOLAD pourront également être intégrés à des environnements tel que Sectra_w, un système collaboratif permettant d'évaluer, de présenter, d'exploiter et de réviser des corpus de traduction automatique (Huynh et al., 2008), par exemple pour proposer l'ajout de dépendances syntaxiques.

Enfin, ACOLAD pourra être proposé dans le cadre de campagne d'évaluation d'analyseurs syntaxiques de dépendances pour aider à fabriquer les analyses références.

Notre plateforme est proposée à la communauté sous licence publique générale limitée GNU (GNU Lesser General Public License). Elle sera prochainement disponible à l'url suivante : https://forge.imag.fr/projects/acolad/.

Références

BLANCHON, H., BOITET, C. AND CHOUMANE, A. (2006). Traduction automatisée fondée sur le dialogue et documents auto-explicatifs: bilan du projet LIDIA. *in TAL*. vol. 47(3):30 p.

BRUNET-MANQUAT F. (2005). Improving dependency analysis by Syntactic parser combination. *Proceedings of IEEE NLP-KE 2005*, Wuhan, China, Oct 30- Nov 1, 2005.

BRUNET-MANQUAT, F. (2004). CRÉATION D'ANALYSEURS DE DÉPENDANCE PAR COMBINAISON D'ANALYSEURS SYNTAXIQUES. THÈSE EN INFORMATIQUE. UNIVERSITÉ JOSEPH FOURIER - GRENOBLE 1. 21 DÉCEMBRE 2004. 169 P.

HOLAN T., KUBON, OLIVA K., PLATEK M. (2000). On complexity of word order, *in TAL*., 41(1), pp. 273-300, Hermès, Paris, France.

HUYNH C.-P., BOITET C. & BLANCHON H. (2008). SECTra_w : an Online Collaborative System for Evaluating, Post-editing and Presenting MT Translation Corpora. *Proc. LREC-08,* Marrakech, 27-31/5/08, ELRA/ELDA, ed., 8 p.

PAROUBEK P., VILLEMONTE DE LA CLERGERIE E., LOISEAU S., VILNAT A., ET FRANCOPOULO G. (2009) The PASSAGE Syntactic Representation, *7th International Workshop on Treebanks and Linguistic Theories* (TLT7, Groningen, January 23-24, 2009)

VALLI, A. VERONIS, J. (1999). Etiquetage grammatical des corpus de parole : problèmes et perspectives. *Revue Française de Linguistique Appliquée, IV(2),* 113-133. (dossier : l'oral spontané)

XAVIER T. (2012). WebAnnotator, an Annotation Tool for Web Pages. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012, to appear)*. Istanbul, Turkey, 2012.