Analyse syntaxique en dépendances et Evaluation

Christine Chardenon
France Télécom Division R&D, TECH/EASY/LN
Christine.Chardenon@francetelecom.com

Mots-clés : analyse syntaxique, dépendances, évaluation
Keywords: syntactical analyzer, dependency grammar, evaluation

Résumé Nous décrivons un analyseur syntaxique et commentons brièvement notre participation à la campagne d'évaluation EASY.

Abstract we describe a syntactical analyzer and we briefly comment our participation to the EASY evaluation campaign.

Introduction

L'équipe Langues Naturelles de France Télécom Division R&D a développé ces dernières années une chaîne de traitement linguistique constituée de modules réalisant des tâches de différents niveaux (lexical, syntaxique, sémantique, ...). Nous avons utilisé cette chaîne pour l'action d'évaluation des analyseurs syntaxiques TECHNOLANGUE/EASY. Nous ferons dans une première partie une brève description des modules nécessaires en amont du module d'analyse syntaxique, qui sera présenté dans une seconde partie. Nous complèterons la présentation de chaque module par une description des adaptations que nous avons faites sur les données utilisées par ce module pendant la phase d'annotation automatique des corpus.

1 Description de la chaîne

Nous distinguons trois étapes principales dans la chaîne de traitement produisant l'analyse syntaxique d'un texte : segmentation du texte, analyse minimale et analyse syntaxique. Nous allons nous intéresser dans cette partie aux deux premières.

Le module de *segmentation* découpe un texte en paragraphes, phrases et segments. Il exploite des données qui déterminent les types de segments et leur associe une description : un segment de type MOT correspond à un ensemble de lettres accentuées ou non, sans espace ni ponctuation, ni chiffres. Une adresse mail est reconnue comme un segment unique. Les segments obtenus sont regroupés en phrases, elles-mêmes regroupées en paragraphes. L'analyse syntaxique peut se faire au niveau phrase ou paragraphe. Durant la phase d'annotation de corpus, les données de segmentation ont été très légèrement adaptées, car

certains segments étaient inutilement découpés (general_elda , adresses de site internet). Nous avons conservé la segmentation en phrases fournie avec les corpus bruts, mais nous avons ensuite appliqué notre segmentation pour retrouver nos types de segments. Certains corpus (general_lemonde) présentait une segmentation en phrase peu pertinente (coupure de phrase après un "M."), le choix de conserver la segmentation en phrases aurait pu être remis en cause.

L'étape d'*analyse minimale* effectue des actions sur chaque segment obtenu lors de l'étape précédente, et ce en fonction de son type. Pour un segment de type MOT, l'action privilégiée est l'analyse lexicale, qui retrouve dans un lexique toutes les interprétations lexicales possibles du texte associé au segment. Le lexique utilisé pour l'action EASY est d'environ 200 000 formes fléchies. Chaque interprétation lexicale permet de créer un ou plusieurs objets appelés *terminaux*. Chaque terminal porte d'une part une catégorie syntaxique principale (*Catep*), d'autre part des informations morpho-syntaxiques, codées sous forme de traits. Par exemple, l'analyse lexicale du mot "livres" donne des terminaux correspondant à des interprétations nominales et verbales. En cas d'échec de l'analyse lexicale, le module contrôle l'application de stratégies de correction. Les types de correction (phonétique, ré-accentuation, analyse morphologique, etc) activées dépendent du corpus traité.

Pour les segments de type autre que MOT, comme les ponctuations, il est possible de créer des terminaux également associés à une description morpho-syntaxique. Par exemple, la virgule génère deux terminaux, l'un de catégorie PONC_GAUCHE et l'autre COORD.

Durant la phase d'annotation, les stratégies de corrections appliquées ont bien été différentes suivant les corpus : ré-accentuation, correction phonétique pour le corpus littéraire par exemple, le faible nombre de fautes d'orthographe dans ce corpus ne justifiant pas l'application de correction typographique. Pour les corpus de type mail, la correction phonétique était efficace, ainsi que la correction morpho-prédictive (prédiction de la catégorie d'un mot en fonction de sa terminaison).

L'étape d'analyse minimale se charge enfin de la reconnaissance des mots composés ou *locutions*. Une locution reconnue donne lieu à la production de terminaux, comme pour les mots simples. Une liste de locutions avait été fournie par les organisateurs de la campagne. Nous avons fait en sorte de compléter notre lexique de locutions quand cela nous a paru nécessaire. Cependant, nous n'avons pas gardé celles qui remettaient en cause les choix linguistiques de notre grammaire (nous traitons "l'un et l'autre" comme une coordination et pas une locution). Nous avons par ailleurs conservé nos locutions pendant l'analyse, il est donc certain que nous avons perdu des relations par rapport aux corpus annoté manuellement (exemple elda/general_elda : "sous traitants" est pour nous une locution).

2 Un analyseur syntaxique basé sur le formalisme des grammaires de dépendance

L'analyseur syntaxique commence par regrouper les terminaux dans des groupes syntaxiques de premier niveau (GS1). Un GS1 rassemble les terminaux issus d'un même segment qui ont la même catégorie principale, celle-ci devenant celle du GS1. Ces terminaux peuvent différer par leurs informations codées sous forme de traits (transitif/intransitif,…). Les terminaux issus de "livres" seraient ainsi répartis entre deux GS1, un de catégorie GN-NC

pour les interprétations nominales, l'autre de catégorie GV-PT pour les interprétations verbales.

L'analyse syntaxique d'une phrase est construite par créations successives de relations entre les GS1. Le processus est bottom-up et se fait par ilôts, générant des analyses partielles de la phrase analysée. Une analyse partielle est associée à un GS1 de tête. En début d'analyse, pour tout GS1, on crée une analyse partielle dont il est la tête. La construction d'une nouvelle analyse se fait par application d'une règle dite de dépendance entre deux analyses partielles déjà construites, ou plus exactement entre les deux GS1 tête de ces analyses. Si une règle s'applique, la nouvelle analyse contient toutes les relations des deux analyses partielles, plus la nouvelle relation créée entre les deux GS1 tête de ces analyses. La règle détermine quel est le GS1 tête de la nouvelle analyse, ainsi que le nom de la relation créée. Dans une analyse partielle donnée, tout GS1 ne peut avoir qu'un père, chaque analyse partielle est donc un arbre.

Les règles de dépendance sont décrites dans des fichiers de grammaire externalisés du module. Leur format est le suivant :

RègleAtt IdentifiantRegle NomRelation **Schéma** (CATEP)* Sens CATEP **CondsPrinc** (Traits*) **CondsDép** (Traits*) **AutresCondConcs** ((Trait*))	**RègleAtt** SUJ-PRN SUJ **Schéma** GV-PT >> PRN-S **CondPrinc** (SY_SUJ/!) **AutresCondConcs** ((P += SY_SUJ) (P/NOMBRE U D/NOMBRE) (P/PERS U D/PERS))

Le premier élément du schéma représente l'ensemble des choix possibles pour la catégorie du GS1 qui sera la tête ou *Principal* de la nouvelle analyse. Le troisième élément représente l'ensemble des choix possibles pour la catégorie du GS1 qui sera le fils ou *Dépendant* de la nouvelle relation. L'élément Sens détermine les positions relatives des deux GS1 (Principal devant ou derrière le Dépendant) dans la phrase. L'ensemble de traits correspondant à l'élément CondsPrinc constitue un filtre de sélection des GS1 candidats à être la tête d'une nouvelle analyse, de même, l'ensemble de traits correspondant à l'élément CondsDép constitue un filtre de sélection des GS1 pouvant être le dépendant d'une nouvelle analyse. Le dernier item est une série de conditions qui doivent être vérifiées par unification entre les ensembles de traits des deux GS1 Principal et Dépendant. Il contient aussi des conclusions à ajouter/retirer aux ensembles de traits résultant de ces unifications. L'application de ces conditions/conclusions se traduit donc par l'affectation d'un nouvel ensemble de traits au GS1 Principal (tête de la nouvelle analyse) et au GS1 Dépendant.

Voici l'analyse de la phrase : "tu livres la porte", dans le format de sortie simplifié utilisé pour générer ensuite la solution au format EASY. A un GS1 sont associés un identifiant (ID), un nom de relation (FONC) s'il n'est pas la tête de l'arbre, et l'identifiant de son Principal (PI).

<GS1 MOT="tu" LEM="tu" CAT="PRN-S" GRA="" ID="0" FONC="SUJ" PI="7" ></GS1>
<GS1 MOT="livres" LEM="livrer" CAT="GV-PT" GRA="" ID="7" ></GS1>
<GS1 MOT="la" LEM="le" CAT="GN-D" GRA="" ID="13" FONC="DET" PI="19" ></GS1>
<GS1 MOT="porte" LEM="Porte" CAT="GN-NC" GRA="" ID="19" FONC="OBJD" PI="7" ></GS1>

Le problème majeur à gérer dans ce type d'analyseurs est l'explosion du nombre d'analyses partielles. En effet, l'attachement de certains éléments de la phrase peut être ambigu en l'absence d'information sémantique levant cette ambiguïté. C'est le cas entre autre pour les attachements de groupes nominaux prépositionnels. Pour des applications à vocabulaire limité (quelques centaines de mots), il est possible d'introduire un contrôle sémantique des attachements. Pour un vocabulaire large, nous ne disposons pas de données sémantiques suffisamment riches pour permettre ce contrôle. L'explosion combinatoire peut être alors contrôlée par applications de diverses stratégies : application prioritaire de certaines règles par rapport à d'autres (par exemple, règles de construction de syntagmes minimaux), priorisation de la construction de relations sous-catégorisées (sujet, objet, etc), limitation de la distance entre un dépendant et sa tête, limitation du nombre d'analyses partielles concurrentes pour un tronçon de phrase, etc. Dans certains cas, l'analyseur produit un ou plusieurs arbres couvrant l'intégralité de la phrase. Parfois, il n'arrive pas à atteindre ce résultat, par manque de couverture de la grammaire pour les énoncés long (corpus du monde), ou parce que les énoncés sont agrammaticaux (corpus de mail). Dans ce cas, il est possible de sélectionner plusieurs arbres syntaxiques successifs qui couvrent la totalité de la phrase, de manière à identifier un maximum de relations syntaxiques (solution en morceaux).
Notre grammaire de dépendance ayant été développée en priorité pour des applications de type requête à un service, elle ne couvrait pas tous les phénomènes de la langue au lancement de la campagne EASY. Nous avons donc fourni un effort pour l'enrichir. Les phénomènes de coordination sont cependant incomplètement gérés, et ce d'autant plus que leur résolution satisfaisante nécessiterait dans certains cas l'exploitation de connaissances sémantiques. Pour des phrases complexes, nous avons donc assez souvent obtenu une solution en morceaux.

3 Conclusion

Pour l'action EASY, nous avons choisi de ne produire que les relations. Pour cela, nous avons fourni des relations entre formes de la phrase car il avait été décidé qu'une relation serait considérée comme valide si les deux formes sur lesquelles elle portait étaient contenues dans les constituants figurant pour cette même relation dans le corpus annoté manuellement. Nous avons par ailleurs choisi de ne pas transformer la sortie actuelle de notre analyseur pour générer des résultats conformes au format requis, mais plutôt d'ajouter un module de transformation de ces sorties. Nous avons développé un programme d'alignement de nos sorties avec les fichiers segmentés fournis par les organisateurs, en tenant compte des divergences portant sur les locutions. En assurant l'alignement, nous avons pu sortir nos relations en les faisant porter sur le numéro de forme correspondant au texte segmenté fourni. En même temps, nous avons renommé nos relations selon les recommandations EASY. Le travail de mise en correspondance de nos 200 noms de relations avec la quinzaine de noms de relations EASY a été assez rapide, mais nous n'avons pas essayé de résoudre finement certains cas de distinction entre compléments du verbe et modifieur du verbe, qui pouvaient être liés en plus à des différences de lexique.

Références

KAHANE S. (2000), Les grammaires de dépendance, *TAL 2000*, Vol. 41 no1, Paris, Hermès.